

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Departamento de Tecnología Electrónica

Grupo de Displays y Aplicaciones Fotónicas

INGENIERÍA INDUSTRIAL

Proyecto Fin de Carrera

RED INALÁMBRICA DE SENSORES DE FIBRA ÓPTICA DE PLÁSTICO CON TECNOLOGÍA ZIGBEE

AUTORA: MERCEDES CHAPARRO DE LA PEÑA

DIRECTORA: CARMEN VÁZQUEZ GARCÍA

Diciembre 2011



Universidad Carlos III de Madrid

Grupo de Displays y Aplicaciones Fotónicas.



Índice General

Índice de Figuras.....	6
Índice de Tablas.....	10
Índice de Gráficas	12
Acrónimos.....	13
Resumen.....	14
Capítulo 1. Introducción y Objetivos	15
1.1. Introducción.....	15
1.2. Aplicaciones de Redes Sensoriales Inalámbricas.....	17
1.3. Objetivos y contenido.....	23
Capítulo 2. Comunicaciones Inalámbricas.....	26
2.1. WiFi.....	28
2.2. Bluetooth.....	29
2.3. ZigBee.....	30
Capítulo 3. Tecnología ZigBee.....	34
3.1. Características de una red sensorial ZigBee.....	36
3.2. Componentes de una Red ZigBee.....	37
3.2.1. El Coordinador.....	38
3.2.2. Los Routers.....	38
3.2.3. End Device.....	38
3.3. Ventajas de las Redes sensoriales con ZigBee.....	38
Capítulo 4. Módulos XBee.....	40
4.1. Modos de operación del módulo XBee.....	42
4.1.1. Modo Recibir/Transmitir.....	42
4.1.2. Modo de bajo consumo.....	42
4.1.3. Modo Comando.....	42



4.1.4.	Modo transparente.	42
4.1.5.	Modo API.	43
4.2.	Direccionamiento de los módulos XBee.	43
4.3.	Tipos de Redes que soporta XBee.	44
Capítulo 5. Descripción del sistema.		48
5.1.	Elementos del Sistema.	50
5.1.1.	Coordinador y Emisores.	50
5.1.2.	Sensores.	53
5.1.3.	Procesamiento de datos.	58
5.2.	Funcionamiento del sistema.	59
5.3.	Configuración del Sistema.	61
Capítulo 6. Pruebas realizadas/ Fases de desarrollo.		62
6.1.	Primera fase de desarrollo del proyecto.	64
6.2.	Segunda fase de desarrollo del proyecto.	68
6.3.	Tercera fase de desarrollo del proyecto.	74
Capítulo 7. <i>Software</i>		82
7.1.	Flujogramas del código fuente.	83
7.2.	Calibración de los módulos Arduino emisores.	86
7.3.	Ampliación de la Red Sensorial.	90
7.4.	Requisitos del <i>software</i> utilizado.	91
Capítulo 8. Diseño final.		92
8.1.	Red de sensores inalámbrica.	96
8.2.	Nodo Receptor.	102
8.3.	Nodos emisores.	103
8.4.	Sensores de Nivel.	105
8.5.	Sensores de Temperatura.	111
Capítulo 9. Conclusiones y Trabajos futuros.		118
Capítulo 10. Bibliografía.		124
Anexo 1. LabWindows. Interfaz gráfica.		129
Anexo 2. X-CTU.		138



Anexo 3. Arduino.....	143
Anexo 4. Pliego de condiciones.....	145
Anexo 5. Presupuesto.....	148
Anexo 6. Código fuente.....	151
I. FASE III DE DESARROLLO.....	151
➤ RED SENSORIAL PUNTO A MULTIPUNTO CON 3 <i>END DEVICE</i> Y UN COORDINADOR.....	151
• PROGRAMA DE CALIBRACIÓN DE MÓDULOS ARDUINO.....	151
• PROGRAMA DE CADA DISPOSITIVO <i>END DEVICE</i>	154
• APLICACIÓN GRÁFICA EN LabWindows.....	161



Índice de Figuras

Figura 1. WSN detección de incendios [Libelium]	18
Figura 2. Equipo situado en cada nodo para detección de incendios, Waspnotes. [Libelium].....	19
Figura 3. WSN en un campo de cultivo. [Libelium]	19
Figura 4. WSN en monitorización de pacientes.[Gom05]	20
Figura 5. WSN en el hogar. [ZigBeeexpo]	21
Figura 6. Logotipo WiFi. [Wi-fi]	28
Figura 7. Logotipo de bluetooth. [Bluetooth].....	29
Figura 8. Logotipo de ZigBee. [ZigBee]	30
Figura 9. Módulos XBee utilizados en este proyecto. [Digi]	41
Figura 10. Conexión Punto a Punto.	44
Figura 11. Conexión Punto a Multipunto.	45
Figura 12. Red <i>NonBeacon</i> con coordinador utilizada en este proyecto.	46
Figura 13. Esquema del sistema implementado.	49
Figura 14. Diagrama de bloques del bloque receptor y del bloque emisor.	50
Figura 15. Placas Arduino Diecimila utilizadas. [Arduino]	50
Figura 16. Esquemático Placa Arduino.[Arduino].....	51
Figura 17. Diagrama de bloques del bloque sensor.	53
Figura 18. Sistema de transmisión por fibra óptica.....	54
Figura 19. Emisor HFBR – 1528. [Datasheet].....	55
Figura 20. IF D91. [Datasheet]	56



Figura 21. Sensor de Temperatura y Sensor de Nivel implementados en cada nodo emisor.	57
Figura 22. Emisor HFBR-1528 utilizado.	58
Figura 23. Esquemático Emisor – POF – Fotodiodo – Arduino.....	60
Figura 24. Esquemático Placa Arduino.[Arduino].....	60
Figura 25. Sistema diseñado en la Fase I.	64
Figura 26. Primera interfaz gráfica diseñada en la Fase I.	65
Figura 27. Segunda interfaz gráfica diseñada en la Fase I.	66
Figura 28. Sistema diseñado en la Fase II.	68
Figura 29. Trama de datos enviada y recibida en modo API.[Digi]	69
Figura 30. Trama de datos enviada por cada emiSor.[Digi]	70
Figura 31. Trama de datos recibida por el coordinador.[Digi]	70
Figura 32. Aplicación realizada al finalizar la segunda fase de desarrollo.	71
Figura 33. Sistema diseñado en la Fase III.	74
Figura 34. Pantalla de la aplicación final diseñada.	75
Figura 35. Fichero data.txt que recoge los datos monitorizados.	76
Figura 36. Plano de los puntos donde se ha comprobado el alcance.	77
Figura 37. Cable USB B-B fabricado para medir la intensidad a la entrada de los Arduinos.....	80
Figura 38. Flujograma de la red punto a punto de la Fase I.	83
Figura 39. Flujograma de la red punto a multipunto de la Fase II.....	84
Figura 40. Flujograma de la red punto a multipunto de la Fase III.....	85
Figura 41. Diagrama de bloques de los componentes de la WSN final.....	93
Figura 42. Montaje final realizado.....	94
Figura 43. Nodo sensor utilizado para verificar la transmisión de datos ante la existencia de paredes.	95
Figura 44. Coordinador unido al PC.....	95



Figura 45. Pantalla perteneciente al diseño final.	100
Figura 46. Fichero data.txt que recoge los datos monitorizados por la aplicación gráfica.	101
Figura 47. Esquemático sensores de nivel.....	106
Figura 48. Sensores POF de nivel.....	107
Figura 49. Esquemático sensores de temperatura.....	111
Figura 50. Sensores POF de temperatura unidos a la placa calefactora.	112
Figura 51. Unidad de control de la placa calefactora.....	113
Figura 52. Creación de un nuevo proyecto en LabWindows.....	130
Figura 53. Creación de un workspace en LabWindows.....	130
Figura 54. Nuevo fichero .c.....	131
Figura 55. Nuevo fichero .uir.	131
Figura 56. Como añadir los ficheros necesarios en LabWindows.	132
Figura 57. Herramientas del editor de LabWindows.....	132
Figura 58. Panel 1 de configuración serie.....	133
Figura 59. Panel 2 de lectura de medidas.	133
Figura 60. Crear paneles, botones, etc. En LabWindows.	134
Figura 61. Configuración de un <i>Command Button</i>	135
Figura 62. Configuración del control <i>Numeric Gauge</i>	136
Figura 63. Configuración del control LED.	137
Figura 64. Configuración control <i>Ring</i>	137
Figura 65. Kit Arduino- Diecimila.[Arduino]	138
Figura 66. Test de verificación del módulo XBee.	139
Figura 67. Direcciones de destino y de cada módulo.....	140
Figura 68. Configuración como <i>End Device</i> o como coordinador y Modo de funcionamiento API.	141



Figura 69. <i>Software</i> de Arduino.....	144
--	-----



Índice de Tablas

Tabla 1. Diferencias entre ZigBee, Bluetooth y WiFi.	32
Tabla 2. Características de la tecnología ZigBee.....	35
Tabla 3. Parámetros de configuración de los módulos XBee.	61
Tabla 4. Resumen de las características de cada Fase de desarrollo.	63
Tabla 5. Características de la red punto a punto desarrollada en la Fase I.....	67
Tabla 6. Características de la red punto a multipunto desarrollada en la Fase II.	73
Tabla 7. Alcance de la red.....	78
Tabla 8. Características de la red punto a multipunto desarrollada en la Fase III.	79
Tabla 9. Calibración Arduino conectado al USB delantero de un PC de sobremesa.....	88
Tabla 10. Calibración Arduino conectado al USB trasero de un PC de sobremesa.	88
Tabla 11. Calibración Arduino conectado al USB de un PC portátil.	89
Tabla 12. Calibración Arduino conectado a la red mediante un cargador.....	89
Tabla 13. Especificaciones de la WSN.....	97
Tabla 14. Comparativa del consumo de los estándares de comunicación.....	97
Tabla 15. Características de las redes diseñadas en cada Fase de trabajo.	99
Tabla 16. Alcance de la red con paredes.	99
Tabla 17. Parámetros de configuración módulo XBee como coordinador.	102
Tabla 18. Especificaciones del nodo receptor.	103
Tabla 19. Parámetros de configuración módulos XBee como dispositivos finales.	104
Tabla 20. Especificaciones de los nodos emisores.	105
Tabla 21. Especificaciones de los sensores de nivel.	107



Tabla 22. Medidas sensores de nivel en el depósito de agua.	108
Tabla 23. Medidas sensores de nivel en el depósito de aceite.	109
Tabla 24. Especificaciones de los sensores de temperatura.	113
Tabla 25. Medidas sensor de temperatura POF sin pulir.	115
Tabla 26. Medidas sensor de temperatura POF pulido.	116
Tabla 27. Especificaciones de la Red Sensorial Inalámbrica.	120
Tabla 28. Valores de Configuración para el modo API de cada módulo XBee.	141
Tabla 29. Valores para una configuración en modo transparente de cada módulo XBee.	142



Índice de Gráficas

Gráfica 1. Representación de las medidas de los sensores de nivel de POF sin pulir.	110
Gráfica 2. Representación de las medidas de los sensores de nivel de POF pulida.	110
Gráfica 3. Representación de las medidas de los sensores de T^a de POF sin pulir.	117
Gráfica 4. Representación de las medidas de los sensores de T^a de POF pulida.	117



Acrónimos

ACRÓNIMO	DESCRIPCIÓN
A/D	Analógico/Digital
API	Aplicación de Interfaz de programación
E/S	Entradas/Salidas
MAC	Control de acceso al medio
mAh	miliamperios hora
POF	Fibra óptica de plástico
RF	Radio Frecuencia
UART	Receptor/ transmisor asíncrono universal
USB	Universal Serial Bus
WLAN	Redes Inalámbricas de Área Local
WPAN	Redes inalámbricas de Área Personal
WSN	Wireless Sensor Network (Red de sensores Inalámbrica)



Resumen.

En este proyecto se aborda un trabajo de investigación en el campo de la Instrumentación electrónica mediante la implementación de una red sensorial inalámbrica (WSN), basada en el estándar IEEE.802.15.4., y el diseño de una interfaz gráfica que permita la monitorización de las medidas recogidas por los nodos sensores en tiempo real.

La red sensorial cuenta con tres nodos sensores y un nodo que hace de estación central encargado de monitorear el nivel discreto de líquido de diferentes depósitos en distintas condiciones y la variación de temperatura de un determinado ambiente. Cada nodo cuenta con dos sensores de fibra óptica de plástico (POF), que recogen los datos de nivel y temperatura, y mediante radiofrecuencia los transmite a la estación central para que sean analizados. La estación central se conecta a un PC por el puerto USB, recoge la información recibida por radiofrecuencia de los diferentes nodos y la representa en la interfaz gráfica diseñada (desarrollada en LabWindows).

Los resultados obtenidos han servido para demostrar la viabilidad de hacer frente a un gran número de sensores ópticos a través de una solución inalámbrica de bajo coste y bajo consumo de energía.

El proyecto se ha desarrollado en el Grupo de Displays y Aplicaciones Fotónicas del Departamento de Tecnología Electrónica gracias a la financiación de los proyectos “Nuevas técnicas de conmutación y sensado en redes ópticas” (TEC2009-1474-C03 03) y “Fotónica Aplicada para la Creación de Tecnologías Ópticas y su Transferencia a Empresas Madrileñas II” FACTOTEM-2 (S2009/ESP-1781).-



Capítulo 1. Introducción y Objetivos

1.1. Introducción.

Hoy en día, las aplicaciones relacionadas con las Redes de Sensores Inalámbricas (WSN) han crecido considerablemente debido a las condiciones de libertad de utilización, de movilidad y flexibilidad que ofrecen estos sistemas, viéndose extendidas a campos como la industria, la sanidad, el medio ambiente y la automatización del hogar y de edificios.

La combinación de la gran variedad de sensores de bajo coste, bajo consumo y reducido tamaño, junto con las tecnologías de transmisión inalámbricas dan lugar a WSN capaces de procesar enormes cantidades de datos [Aak+06] . Siendo la comunicación inalámbrica un tipo de comunicación en la que no se utiliza un medio físico de propagación (*wireless*, sin cables) ya que utiliza ondas electromagnéticas de baja potencia y una banda específica para transmitir entre dispositivos. [Gra07]

Una WSN está formada por numerosos dispositivos (llamados nodos) que utilizan sensores distribuidos espacialmente en diferentes puntos y comunicándose entre sí de manera inalámbrica, se encargan de analizar información como temperatura, nivel, presión, etc. en zonas específicas. La información recogida puede ser enviada a un nodo que actúe de pasarela o a un nodo central de comunicación.

Cada nodo cuenta con un dispositivo autónomo formado por un microcontrolador, una batería, un radio-transceptor y un elemento sensor. La capacidad de procesamiento dependerá del tipo de microprocesador que se emplee, la fuente de alimentación limitará la vida útil de la red, la comunicación se realizará mediante un transceptor que hace las funciones de transmisión y recepción y por último, el elemento sensor se encargará de recoger la información de interés (un sensor es un dispositivo capaz de transformar magnitudes físicas o químicas en magnitudes eléctricas, existiendo sensores para medir parámetros tales como desplazamiento, temperatura, presión, fuerza, humedad relativa, aceleración, etc.). [Gra07]

Los sensores inalámbricos presentan grandes ventajas tales como [Gas10] :

- Costes reducidos y facilidad de instalación gracias al pequeño tamaño de los nodos, al reducido consumo y a la existencia de tecnologías abiertas que da la posibilidad de operación entre dispositivos de diferentes fabricantes.
- Amplia cobertura debida a la no utilización de cables, ya que se puede llegar a sitios donde los cables no son accesibles.
- Bajo consumo de potencia gracias a los modos *sleep*, *wake up* y *active*, que dotan al sistema de una larga autonomía.
- Precisión y frecuencia de mediciones gracias al amplio número de nodos que puede tener la red.

Pero a su vez cuentan con restricciones como memoria, capacidad de cómputo y probabilidad de fallo al formar parte de redes desatendidas (reducido con el monitoreo de la red).



Las aplicaciones de las Redes Sensoriales Inalámbricas incluyen la supervisión, seguimiento y control.

Algunos de sus usos son: **[Ser07]**

- Control de parámetros en la agricultura, temperatura, humedad, pH, etc.
- Monitorización del medio ambiente, observación del suelo o agua, detección de incendios.
- Monitorización de un hábitat (para determinar la población y comportamiento de animales y plantas).
- Sensorización de edificios “inteligentes”, controlando iluminación, apertura de puertas, climatización, etc.
- Control de tráfico, agilizando el aparcamiento de vehículos en diferentes garajes de grandes superficies gracias al aviso de disponibilidad y ocupación de plazas.
- Asistencia militar o civil, control de la trayectoria de los misiles, detección de minas, etc.
- Control médico, transmitiendo datos de equipos de medida (pulsímetros, termómetros,...) para monitorizarlos de forma remota e instantánea.

En el apartado 1.2 se dará una visión más detallada de algunas de las aplicaciones de las WSN.

1.2. Aplicaciones de Redes Sensoriales Inalámbricas.

Actualmente las Redes Sensoriales Inalámbricas son capaces de dar soporte a un amplio abanico de aplicaciones en campos como el medio ambiente, la agricultura, la salud y el hogar.

La reducción de costes gracias al ahorro en cableado de comunicación, la mejora de las redes de comunicaciones con la aparición de nuevos protocolos de comunicación de bajo consumo y baja transferencia de datos, y el reducido tamaño de los nodos sensores han contribuido al desarrollo e implantación de estos sistemas.

- En el caso del medio ambiente, las grandes dimensiones y la dificultad de acceso a determinadas zonas de Parques Naturales dificultan las acciones de cuidado y mantenimiento de estos lugares. Si a eso se le suma la existencia de animales y especies vegetales que hay que preservar, se complica más la tarea de supervisión. Las WSN han permitido la utilización de un método no intrusivo capaz de monitorear parámetros como conteo de animales, humedad, temperatura, cantidad de monóxido de carbono, etc. La inexistencia de cables de comunicación y el pequeño tamaño de los nodos sensores reducen el impacto visual ya que pueden disimularse en el entorno con facilidad mientras transfieren los datos recogidos a la estación central, véase la Figura 1.



Figura 1. WSN detección de incendios [Libelium] .

Existe un plan de vigilancia y seguimiento ambiental para la detección de incendios forestales, integrado en una extensión de 210 hectáreas pertenecientes a las Comunidades Autónomas de Galicia y Asturias, donde se han desplegado 90 nodos, con cuatro elementos sensores por nodo, en puntos estratégicos. Cada 5 minutos se realiza una medida de temperatura, de humedad relativa, de cantidad de monóxido de carbono y de dióxido de carbono y los datos son enviados a un *router* multiprotocolo mediante tecnología ZigBee, encargándose este último de transmitirlos al Centro de Control vía WiFi **[Libelium]** . Este sistema ha sido desarrollado por la compañía DIMAP-FactorLink con los productos de Libelium **[Dimap]** , un ejemplo de uno de los nodos utilizado se puede ver en la Figura 2, con

el propósito de obtener una rápida detección de posibles focos de incendio que puedan ser extinguidos antes de su propagación y así proteger la flora y la fauna de la zona.



Figura 2. Equipo situado en cada nodo para detección de incendios, Waspnotes. [Libelium]

- Poder monitorizar parámetros como la temperatura, el clima, la humedad o el pH en la agricultura permite una mejora en la calidad de los productos y una optimización de recursos. Con la monitorización de dichos parámetros mediante una WSN como la que se muestra en la Figura 3 se logra gestionar los cultivos y las cosechas consiguiendo una reducción en el uso de pesticidas (conociendo la zona donde es necesario en cada momento) y en el consumo de agua (zonas de riego específicas) y una detección rápida de situaciones adversas que permite llevar a cabo antes un protocolo de actuación.



Figura 3. WSN en un campo de cultivo. [Libelium]

La empresa Libelium cuenta con un Wasp mote (nombre del dispositivo sensorial creado por la empresa para agricultura de precisión) capaz de monitorizar de forma remota un máximo de catorce parámetros medioambientales en una WSN como: temperatura del aire, humedad relativa del aire, temperatura del suelo, humedad del suelo, humectación de la hoja, presión atmosférica, radiación solar, diámetro de tronco/tallo/fruto, velocidad del viento, dirección de viento y precipitación. Los datos adquiridos son enviados mediante el protocolo ZigBee inalámbricamente hasta una distancia de 12 kilómetros pudiendo así medir la efectividad del riego, el crecimiento de la cosecha, las condiciones climáticas, prevenir las heladas, detectar la necesidad de riego según el nivel de humedad de la tierra, etc.

[Libelium]

- En el ámbito de la salud, algunas de las posibilidades que brindan las WSN son: la monitorización de pacientes, como se ve en la Figura 4, la administración de medicinas, el seguimiento de la enfermedad y la localización de pacientes. Estas tecnologías proporcionan a los pacientes mayor confianza y mejoran su calidad de vida ya que se puede dar una detección más rápida de estados anormales y una supervisión continua.



Figura 4. WSN en monitorización de pacientes.[Gom05]

Científicos de la Universidad Carlos III de Madrid han desarrollado una camiseta “inteligente” que monitoriza el cuerpo humano de quien la lleva puesta registrando su temperatura corporal, la frecuencia cardiaca, la posición del paciente y la postura de un modo no intrusivo. La información captada se envía de forma inalámbrica mediante ZigBee a un sistema gestor instalado en el hospital.

El prototipo ha sido desarrollado en el marco del proyecto LOBIN y ha sido probado por pacientes de la Unidad de Cardiología del Hospital de La Paz en Madrid. **[Lop+10]**

- El control automático en el hogar y en edificios es un campo que puede resultar muy interesante para la implementación de las WSN. Sus características de comunicación inalámbrica, transmisión de variables de entorno y posibilidades de actuación sobre determinados elementos las hacen una alternativa clara y eficaz en aplicaciones domóticas.

El control de iluminación, climatización, de aparatos electrónicos, la detección de gases, etc. todo ello mediante dispositivos sensores asociados a una red inalámbrica hacen que las WSN jueguen un papel importante en el diseño del hogar inteligente como se puede ver representado en la Figura 5, proporcionando confort, seguridad y ahorro económico/energético al usuario.

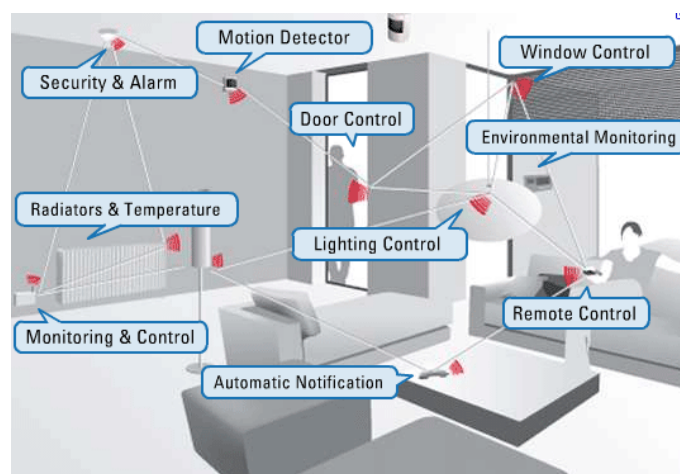


Figura 5. WSN en el hogar. [ZigBeeexpo]



El hotel Las Américas en Santiago de Cuba dispone de una instalación solar térmica con colectores solares que concentran la energía solar incidente en ellos para después calentar el agua potable proveniente de la red hidráulica.

Se utiliza la tecnología ZigBee para monitorear el sistema de calentamiento de agua con energía solar. Se controla la temperatura del agua para las habitaciones a través de la manipulación del caudal de agua que recircula entre los calentadores solares y el tanque de almacenamiento.

La instalación se encarga de hacer circular el agua caliente por las habitaciones cuando la temperatura del agua de las habitaciones está muy por debajo de la del tanque que es analizada mediante sensores **[Pul+10]** . Con esta instalación se consigue un ahorro económico y energético.

Existen diferentes proyectos dedicados a la implementación de redes domóticas con el estándar ZigBee. La versatilidad de los módulos ZigBee para realizar un control domótico en un hogar, el fácil montaje dentro del mismo de una WSN que controla la iluminación, las tomas eléctricas, las persianas y las electroválvulas del gas de la vivienda sin necesidad de realizar ninguna obra, hacen viable la existencia de este tipo de redes y la actual investigación sobre los avances de esta tecnología. **[Rei+]**

Otro punto a destacar es el concepto de medición inteligente, donde se están desarrollando diferentes proyectos por parte de los principales operadores eléctricos en España. Se busca tener una lectura en tiempo real de los contadores del consumo eléctrico, dando lugar a la facturación automática. Así se consigue mantener informado al consumidor-productor de energía de los valores actuales del flujo energético y cuantificar instantáneamente el estado de la red de distribución en el lado del consumidor. **[Enerzuul]**



1.3. Objetivos y contenido.

El presente proyecto se ha realizado en el Grupo de Displays y Aplicaciones Fotónicas del Departamento de Tecnología Electrónica gracias a la financiación del Ministerio de Ciencia e Innovación a través del proyecto “Nuevas técnicas de conmutación y sensado en redes ópticas” (TEC2009-1474-C03 03) y de la Comunidad de Madrid a través del proyecto “Fotónica Aplicada para la Creación de Tecnologías Ópticas y su Transferencia a Empresas Madrileñas II” FACTOTEM-2 (S2009/ESP-1781).-

Tras estudiar las diferentes aplicaciones de las WSN, y llegar al punto de la implantación de este tipo de redes en el hogar, surge el interés por parte del Grupo de Displays y Aplicaciones Fotónicas de la Universidad Carlos III de Madrid de abrir una línea de investigación basada en las Redes Sensoriales Inalámbricas relacionadas con los sensores de fibra óptica como detectores de nivel de líquido de un depósito de combustible en una vivienda y la temperatura de dicho lugar.

Por ello, en el siguiente proyecto se pretende desarrollar una red sensorial inalámbrica de bajo coste, bajo consumo de energía y capaz de albergar un gran número de dispositivos sensores, con una interfaz gráfica encargada de monitorear los resultados.

La medida del nivel de líquido y de temperatura se llevará a cabo mediante sensores de fibra óptica de plástico (POF), ya que una de las aplicaciones más interesantes de la fibra óptica es su uso como medio de sensado. Su bajo coste, su insensibilidad ante interferencias electromagnéticas y su seguridad intrínseca (por ellas no circula corriente eléctrica) les hace tener ciertas ventajas con respecto a los sensores eléctricos.

De este modo, la fibra óptica se puede utilizar para medir diferentes parámetros como pueden ser tensión, temperatura, presión, nivel de líquido, etc. y se ha demostrado la capacidad de los sensores basados en POFs para medir los niveles de líquidos en ambientes hostiles, inflamables y de difícil acceso.



Se busca diseñar una red multipunto formada por distintos nodos capaces de sensor temperatura, y el nivel de líquido de varios depósitos, con dispositivos de bajo consumo y un alcance no muy grande debido a la ubicación de la red en una vivienda.

De forma más específica, los objetivos principales que se buscan con el presente proyecto son:

- Trabajar en un medio inalámbrico que proporcione movilidad y flexibilidad al sistema de medida.
- Uso de técnicas ópticas que permitan la medida segura en entornos difíciles (riesgo explosión, exceso de radiación, biocompatibles, etc.)
- Controlar los sensores de fibra óptica con dispositivos inalámbricos.
- Sistema de bajo consumo.
- Diseñar una aplicación gráfica de fácil y sencillo manejo a la vez que proporcione una rapidez de lectura de los datos recogidos, es decir, que sea lo más visual posible.

En cuanto al contenido del documento, a lo largo del mismo se describirá el diseño, desarrollo e implementación de las aplicaciones y consideraciones necesarias para la realización de este proyecto.

El Capítulo 2 expone una descripción de las distintas comunicaciones inalámbricas barajadas para ser utilizadas en este proyecto.

En el Capítulo 3 se tratan las características de la Tecnología ZigBee, los diferentes componentes de una red ZigBee y las ventajas de realizar una red sensorial con este tipo de comunicación.

A lo largo del Capítulo 4 se aborda la parte correspondiente a los módulos de radiofrecuencia XBee, para qué sirven, cómo se configuran, etc.

En el Capítulo 5 se describe el sistema diseñado, su funcionamiento y sus características.

El Capítulo 6 expone las fases en las que se han dividido las diferentes pruebas realizadas durante la ejecución de este proyecto para llegar al diseño final obtenido.



En el Capítulo 7 se describe el *software* utilizado en este proyecto.

A lo largo del capítulo 8 y del capítulo 9 se muestra el diseño final obtenido, los resultados y las conclusiones una vez finalizado el proyecto.

El capítulo 10 hace referencia a la bibliografía del proyecto.

En los anexos se incluye la siguiente información sobre el diseño y desarrollo del sistema:

- Anexo 1. Donde se explica cómo se configura y utiliza LabWindows *software*.
- Anexo 2. Donde se explica cómo se configura y utiliza XCTU *software*.
- Anexo 3. Donde se explica cómo se configura y utiliza Arduino *software*.
- Anexo 4. Pliego de Condiciones.
- Anexo 5. Presupuesto.
- Anexo 6. Con el código fuente desarrollado.



Capítulo 2. Comunicaciones Inalámbricas.

Una red inalámbrica es aquella cuyo medio de transmisión son las ondas electromagnéticas, siendo la transmisión y la recepción a través de antenas.

Las ventajas más destacadas en este tipo de comunicación son:

- Rápida instalación de la red sin la necesidad de usar cableado.
- Permiten la movilidad y tienen menos costes de mantenimiento que una red convencional.

[Manual-Wifi]



El amplio crecimiento de las tecnologías inalámbricas ha permitido simplificar bastante las tareas cotidianas, mejorando el bienestar de las personas.

Dependiendo del alcance, se pueden clasificar las diferentes tecnologías inalámbricas en:

- Redes inalámbricas de Área Personal (WPAN). Son redes cuya finalidad es la comunicación entre dispositivos cercanos al punto de acceso debido a que tienen un alcance pequeño. Diferentes estándares desarrollados para las comunicaciones inalámbricas de tipo WPAN son bluetooth y ZigBee.

Para muchos, bluetooth y ZigBee están todas agrupadas en la misma categoría, cuando en realidad representan muy diferentes etapas de desarrollo y ofrecen varios niveles de funcionalidad.

- Redes Inalámbricas de Área Local (WLAN). Tienen un alcance superior a las WPAN, superando un mayor número de obstáculos. Son muy populares para acceder a Internet, y el estándar más conocido en este tipo de redes es el WiFi.

A continuación se va a realizar un análisis de cada tipo de estándar para hacer una elección de cuál es el que mejor cubre las necesidades de este proyecto:

- Baja transmisión de datos.
- Bajo consumo.
- Bajo precio.
- Pequeño tamaño de los módulos.

2.1. WiFi.

WiFi es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE. 802.11. Se creó para ser utilizada en redes locales inalámbricas, pero es frecuente que en la actualidad se utilice para el acceso a Internet.



Figura 6. Logotipo WiFi. [Wi-fi]

WiFi es una marca de la WiFi Alliance (anteriormente la *Wireless Ethernet Compatibility Alliance*), su logotipo se muestra en la Figura 6, la organización comercial que prueba y certifica que los equipos cumplen los estándares IEEE 802.11x. [Wi-fi]

WiFi o WLAN es una red que requiere la actividad casi ininterrumpida de los dispositivos en la red. La ventaja de este estándar es la cantidad tremenda de datos que se pueden transferir de un punto a multi-puntos, pero se puede ver que la corriente en transmisión es alta.

Funciona a una frecuencia de 2.4 GHz, al igual que la tecnología inalámbrica bluetooth, sin embargo, en la versión 1.2 y mayores del estándar bluetooth se ha actualizado su especificación para que no haya interferencias en la utilización simultánea de ambas tecnologías.

Este tipo de sistemas requieren de unos 0.6 Watios de potencia. La máxima potencia de la señal que se transmite puede ser superior a 20 dBm.

Uno de los problemas más graves a los cuales se enfrenta actualmente la tecnología WiFi es la seguridad. Un muy elevado porcentaje de redes son instaladas por administradores de sistemas y redes por su simplicidad de implementación sin tener en consideración la seguridad y, por tanto,

convirtiendo sus redes en redes abiertas, sin proteger la información que por ellas circula. **[Manual-Wifi]**

2.2. Bluetooth.

Fundamentalmente, bluetooth, cuyo logotipo aparece en la Figura 7, vendría a ser el nombre común de la especificación industrial IEEE 802.15.1, que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura, globalmente y sin licencia de corto rango.



Figura 7. Logotipo de bluetooth. [Bluetooth]

Los principales objetivos que se pretende conseguir con esta norma son: **[Bluetooth]**

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Bluetooth proporciona una vía de interconexión inalámbrica entre diversos aparatos que tengan dentro de sí esta tecnología, como teléfonos, cámaras, ordenadores portátiles, impresoras, etc. El

alcance que logran tener estos dispositivos es de 10 metros. Para mejorar la comunicación es recomendable que nada físico (como una pared) se interponga.

La especificación de bluetooth define un canal de comunicación de máximo 720Kb/seg con rango óptimo de 10 metros.

La frecuencia de radio con la que trabaja está en el rango de 2.4 a 2.48GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos/seg. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1MHz; esto permite dar seguridad y robustez.

La potencia de salida para transmitir a una distancia máxima de 10 metros es de 0dBm (1 mW), mientras que la versión de largo alcance transmite entre -30 y 20dBm (100 mW).

2.3. ZigBee.

Esta tecnología es la más nueva en este ambiente. ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal y se representa con el logotipo de la Figura 8.



Figura 8. Logotipo de ZigBee. [ZigBee]

ZigBee está diseñado para operaciones de baja potencia. Un dispositivo ZigBee puede dejarse en *standby* por un periodo largo de tiempo sin necesidad de volver a cargar la batería de ese dispositivo. Esto elimina la necesidad del operador de recargar la batería frecuentemente.



ZigBee también ofrece mayor rango que bluetooth. Está diseñado para dar servicio a dispositivos con baja transmisión de datos frente a dispositivos que requieren banda ancha para transmitir video y gráficos. **[ZigBee]**

ZigBee es similar al bluetooth pero con algunas diferencias:

- Menor consumo eléctrico.
- Menor velocidad de transferencia.

Las comunicaciones ZigBee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza FHSS (*Frequency hopping*), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles.

El alcance depende de la potencia de emisión del dispositivo así como el tipo de antenas utilizadas (cerámicas, dipolos, etc.). El alcance normal con antena dipolo en visión directa suele ser aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m.

La velocidad de transmisión de datos de una red ZigBee es de hasta 256kbps y su consumo es de aproximadamente unos 120 mW.

Por último decir que una red ZigBee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. La realidad es menor, siendo, de todas formas, de miles de equipos. **[Blagelectronica]**

Una vez analizados los detalles principales de los estándares WiFi, bluetooth y ZigBee, se ha realizado un cuadro resumen, recogido en la Tabla 1, donde se puede ver más detalladamente las características de cada sistema.

Los parámetros mostrados en la Tabla 1 son el estándar del MAC, la máxima velocidad de transmisión, el consumo de corriente típica en transmisión en *standby*, los requisitos de memoria de programa para un dispositivo típico, aplicaciones y opciones de conexión de la red, y así ver cual

cumple con las especificaciones que se necesitan en este proyecto: bajo consumo, bajo alcance, y bajo coste.

COMPARATIVA PRINCIPALES ESTÁNDARES DE COMUNICACIÓN		
ZigBee	Bluetooth	WiFi
Estándar 802.15.4.	Estándar 802.15.1.	Estándar 802.11.
250 kbps	1 Mbps	Hasta 54 Mbps
TX 35 mA	TX 40 mA	TX > 400 mA
Standby 3µA	Standby 200 µA	Standby 20 mA
32-60 KB memoria	100+ KB memoria	100+KB memoria
Iluminación, sensores, control remoto...	Telecomunicaciones, audio...	Internet....
Red en malla, punto a punto o punto a multipunto	Punto a multipunto	Punto a multipunto

Tabla 1. Diferencias entre ZigBee, Bluetooth y WiFi.

De la comparativa entre los diferentes estándares de comunicación más populares que comparten la banda de 2,4 GHz sin licencia que muestra la Tabla 1, solamente ZigBee brinda la flexibilidad de la conexión de redes en malla.

También se pueden ver los reducidos requisitos de memoria de programa de ZigBee. Las aplicaciones ZigBee son típicamente muy simples. El consumo de potencia es reducido por el hecho de que los dispositivos “end point” de ZigBee pueden “dormir” mientras que se mantienen asociados a la red.

Uno de los puntos clave de esta tabla es mostrar que los estándares inalámbricos están basados en lo que se llaman “modelos de uso” o “aplicaciones”. Ningún estándar cubre todos los requerimientos de todos los “modelos de uso”. Los diseñadores deben escoger el estándar que cubre mejor sus requisitos de aplicación.



Para elegir el tipo de tecnología inalámbrica que se adapta a nuestras necesidades nos hemos fijado en unos parámetros específicos de cada estándar como son: cobertura, consumo, precio, frecuencia de trabajo y el protocolo de comunicación.

Tras este análisis, se ha decidido usar la tecnología ZigBee mediante el uso de módulos XBee por diversos motivos:

- Estos módulos sirven para hacer redes sensoriales.
- Tienen una tasa de transferencia baja, de Kbs, el consumo es reducido y su funcionamiento es por radio frecuencia (2,4 GHz).
- Posee E/S digitales.
- Convertidor A/D con 4 entradas analógicas ó 4 E/S digitales.
- Puede incorporar una clave para las comunicaciones.
- Con él se pueden unir redes.
- Alcance de 100 metros en campo abierto y de 30 metros con paredes.
- Ancho de banda de Kbs.
- Para su uso, sólo es necesario un PC, el cable de comunicación con el PC, y la alimentación del módulo que vaya unido al emisor.
- *Software* gratuito. DIGI es el proveedor del *software*, el nombre de este *software* es X-CTU, que permite configurar los parámetros del módulo y es como un hyperterminal.
- En este proyecto se usará un Kit formado por un Arduino Diecimila y un módulo de comunicación XBee, que será el encargado de dar la conexión al PC.

A continuación, en el siguiente capítulo, se explicará de forma más detallada en qué consiste la comunicación ZigBee.



Capítulo 3. Tecnología ZigBee.

El término ZigBee describe un protocolo inalámbrico normalizado, de bajo coste, para la conexión de una Red de Área Personal o WPAN. Este protocolo surge de la alianza de 25 empresas con el objetivo de ayudar al desarrollo de la tecnología inalámbrica.

Las características más importantes del estándar IEEE 802.15.4 son la flexibilidad de la red, el bajo coste y el bajo consumo de energía; este estándar se puede utilizar para muchas aplicaciones domóticas e industriales, donde se requiere una baja tasa de transmisión de datos.

Los sistemas domóticos pueden utilizar una red cableada o inalámbrica. Los principales sistemas cableados son las líneas telefónicas, módems por cable y líneas de transmisión de energía eléctrica. Cada uno de ellos ofrece ventajas y desventajas que dependen principalmente de su capacidad de ancho de banda, instalación, mantenimiento y costo, entre otros. Las redes inalámbricas implican un gran intercambio de información con un mínimo de esfuerzo de instalación.

ZigBee se ha implementado en la banda mundial de 2.4GHz, sin necesidad de licencia, las frecuencias 868/900 MHz y 2.4 GHz tienen buena penetración tanto a través de paredes y de techos, pero tienen un rango limitado. La limitación de rango es realmente deseable para reducir las interferencias.

El *hardware* y las redes ZigBee deben permitir una transferencia de datos de 10 a 115 kbps. El *hardware* ZigBee puede comunicarse sobre un rango entre 10 a 75 metros. Un *hardware* típico a 2.4 GHz presenta una distancia de trabajo de hasta 30 metros dentro de un edificio y más de 100 metros en campo abierto. **[ZigBee]**

Se pueden poner hasta 100 redes de ZigBee funcionando conjuntamente. Los dispositivos de la red, sensores inalámbricos, monitores o controladores pueden funcionar hasta 2 años con baterías del tipo AA y AAA. En la Tabla 2 se describen las principales características de una red ZigBee:

CARACTERÍSTICAS DE UNA RED ZigBee	
BANDA DE FRECUENCIA Y RANGO DE TRANSMISIÓN DE DATOS	868 MHz → 20 Kb/s 915 MHz → 40 Kb/s 2,4 GHz → 250 Kb/s
ALCANCE	10 – 30 m
LATENCIA	Por debajo de 15 ms
CANALES	868/915 MHz → 11 canales 2,4 GHz → 16 canales
MODOS DE DIRECCIONAMIENTO	Todos los chips tienen 64 bits IEEE de direccionamiento
CANAL DE ACCESO	CSMA-CA
SEGURIDAD	128 AES
RED	Hasta 2^{64} dispositivos
RANGO DE TEMPERATURA	-40 a +85 °C

Tabla 2. Características de la tecnología ZigBee.



Algunas de las características más destacadas de esta tecnología inalámbrica, que se muestran en la Tabla 2 son:

- Diferentes bandas de trabajo: 2.4 GHz (16 Canales), 915 MHz (10 Canales), 868 MHz (1 Canal).
- Tasas de transferencia: 250 Kb/s, 40 Kb/s, 20 Kb/s.
- Topologías: estrella y p2p (punto-a-punto).
- Direccionamiento MAC recortado (16 bits) y extendido (64 bits).
- Bajo consumo energético.
- Gran densidad de nodos por red.
- Radio medio de alcance: 50 m (hasta 500 m, dependiendo del entorno).
- Direccionamiento a nivel de red (16 bits).
- Soporte para enrutamiento de paquetes.
- Permite topología de malla, gracias a las posibilidades de enrutamiento.
- Dispositivos FFD (Coordinador, *router* y dispositivo final) y RFD (dispositivo final).

3.1. Características de una red sensorial ZigBee.

Las redes sensoriales están formadas por un conjunto de nodos con un sensor acoplado que pueden monitorizar parámetros del entorno. Algunos de estos valores son: temperatura, humedad, humo, luz, sonido, movimiento, etc.



Los nodos tienen la capacidad de comunicarse entre sí con el protocolo ZigBee y retransmitir la información adquirida a través de la red hasta un punto de control que registre los valores observados e incluso tome decisiones consecuentemente. Además, son los propios nodos los que se organizan automáticamente cada cierto tiempo para ver cuáles son las rutas de comunicación disponibles. Esto permite que los nodos puedan ser cambiados de lugar para monitorizar un área distinta en un determinado momento. Por otra parte, también hace que la red sea escalable.

La escalabilidad es muy importante dada la naturaleza de los escenarios en los que pueden ser instaladas. Por un lado, hace posible incrementar el área monitorizada simplemente añadiendo nodos y sin necesidad de configuración alguna, ya que serán incorporados a la red dinámicamente. Por otra parte, el hecho de que uno de ellos desaparezca no perjudica el funcionamiento global de la red ya que la información puede seguir siendo transmitida por caminos alternativos. Esto último representa una gran ventaja frente a diseños centralizados ya que proporciona un servicio de alta disponibilidad. **[Libelium]**

La alimentación de la red sensorial se realiza de manera autónoma a través de baterías, lo que permite su instalación en sitios en los que no haya corriente eléctrica como campos de cultivo, bosques, invernaderos, jardines, etc.

Los nodos de la red sensorial no necesitan emitir continuamente, basta con que tomen una muestra cada cierto periodo de tiempo y permanezcan en estado latente el resto del tiempo, optimizando así el consumo energético. De esta manera, se puede conseguir alimentar cada nodo durante meses con sólo dos pilas (AA).

3.2. Componentes de una Red ZigBee.

Una red ZigBee la forman básicamente 3 tipos de elementos. Un único dispositivo coordinador, dispositivos *Routers* y dispositivos finales (*end points* o *end devices*).



3.2.1. El Coordinador.

Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el coordinador puede formar una red, permitiendo unirse a él a dispositivos *Routers* y *End Points*. Una vez formada la red, el coordinador hace las funciones de *Router*, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

3.2.2. Los *Routers*.

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información. Lógicamente un *router* debe unirse a una red ZigBee antes de poder actuar como *Router* retransmitiendo paquetes de otros *routers* o de *End points*.

3.2.3. *End Device*.

Los dispositivos finales son los encargados de recoger la información de los sensores que a ellos van unidos y transmitirlas al medio para que estos datos sean gestionados por el coordinador.

Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

3.3. Ventajas de las Redes sensoriales con ZigBee.

Las ventajas que se pueden obtener de las redes sensoriales con ZigBee son:

- Escalabilidad. La capacidad de auto-organización de la red permite que la monitorización de mayores espacios pueda hacerse fácilmente añadiendo nuevos nodos. Si se añaden nuevos comunicadores, éstos son reconocidos automáticamente por el resto, los cuales actualizarán sus tablas de rutas en consecuencia.
- Alta disponibilidad. La misma característica de auto-organización permite que el área de monitorización también pueda reducirse. En caso de que un nodo deje de funcionar, el resto



de valores pueden seguir siendo transmitidos por caminos alternativos gracias al algoritmo de enrutamiento dinámico, que se encarga de actualizar las rutas. De esta manera, se garantiza siempre el funcionamiento del resto de la red, al contrario de lo que sucedería con un modelo de comunicación centralizado.

- Fácil implantación. El hecho de que la comunicación sea inalámbrica, eliminando así la necesidad de trasladar cables de red, y que la alimentación se realice por medio de baterías, hace que la instalación sea fácil y posible en cualquier entorno, abriendo así un gran abanico de aplicaciones que pueden beneficiarse de esta tecnología.



Capítulo 4. Módulos XBee.

En este capítulo se va a hablar acerca de los módulos XBee y XBee PRO, dos integrantes de la familia DIGI RF, anteriormente conocida como MaxStream. Se trata de módulos integrados e inteligentes, mediante los cuales es posible operar en redes 802.15.4, ZigBee.

Los módulos XBee son transceptores de radio frecuencia que trabajan en la banda de 2.4 GHz con protocolo de comunicación 802.15.4.

El módulo XBee, representado en la Figura 9, presenta una potencia de salida de 1mW, y la sensibilidad del receptor es de -92dBm. Esto le permite operar hasta a unos 100m en espacios abiertos, y hasta 30m en espacios urbanos.

El módulo XBee PRO presenta una potencia de salida de 60mW (18dBm), y la sensibilidad del receptor es de -100dBm. Esto le permite operar a más de 1km en espacios abiertos, y hasta 300m en espacios urbanos.

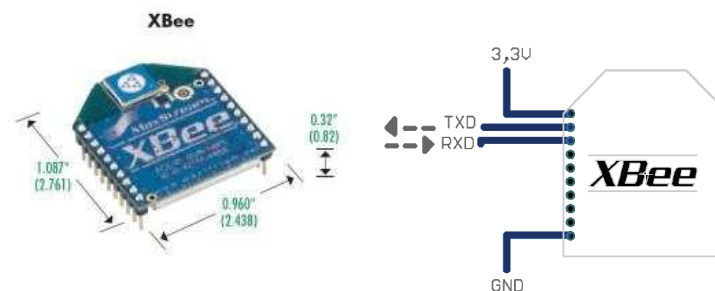


Figura 9. Módulos XBee utilizados en este proyecto. [Digi]

La característica fundamental que ha destacado a los productos MaxStream es la sensibilidad de sus receptores. En vez de elevar la potencia del transmisor, con un aumento de consumo, para lograr mayor alcance, los productos MaxStream incorporan receptores más sensibles, lo que les permite lograr un mayor alcance que otros productos similares, manteniendo un consumo reducido.

La corriente de operación de estos dispositivos ronda los 50mA, tanto para transmisión (215mA a máxima potencia para el XBee PRO) como para recepción, mientras que en el modo de bajo consumo se reduce a tan sólo 10µA. La tensión de operación es 3 a 3,3V.

Las opciones de antena son: antena chip integrada en el módulo, antena de cuarto de onda en el módulo, conector MMCX para antena externa al módulo

El diseño exterior está basado en un formato de algo menos de 27x25mm, la conexión a la PCB se realiza mediante dos conectores de 10 pines de paso 2mm separados 22mm entre sí.



4.1. Modos de operación del módulo XBee.

Los módulos XBee tienen 5 modos diferentes de operar:

4.1.1. Modo Recibir/Transmitir.

Se encuentra en estos modos cuando el módulo recibe algún paquete RF a través de la antena (modo *Receive*) o cuando se envía información serial al *buffer* del pin 3 (UART *Data in*) que luego será transmitida (modo *Transmit*).

La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida por el módulo durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita.

4.1.2. Modo de bajo consumo.

El modo de Bajo Consumo o modo *Sleep*, corresponde cuando el módulo entra en un estado de bajo consumo de energía. Esto depende de la configuración en la que se encuentra. Se debe indicar que el ahorro de energía depende enormemente del voltaje de alimentación.

4.1.3. Modo Comando.

Este modo permite ingresar comandos AT al módulo XBee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

4.1.4. Modo transparente.

En el modo transparente, el módulo envía al remoto configurado como destinatario los mensajes que recibe por su puerto serie, y presenta en éste los mensajes que recibe del módulo remoto. Los cambios de configuración se realizan escapando a modo comando e ingresando comandos AT.



4.1.5. Modo API.

En el modo API, especialmente recomendado para el módulo que debe cumplir la función de coordinador, tanto comandos como mensajes y respuestas viajan dentro de un *framing* documentado, permitiendo mandar y recibir mensajes a y de múltiples remotos de forma sencilla, en una única interfaz serie.

4.2. Direccionamiento de los módulos XBee.

Poner en marcha por primera vez uno de estos módulos es sencillo, un programa gratuito (XCTU) nos permite configurarlo de forma clara y simple, mediante una interfaz que nos deja ver los comandos AT sin necesidad de recordarlos para poder usarlos.

El direccionamiento de los módulos XBee puede ser de 16 ó de 64 bits. Si se utiliza el direccionamiento de 64 bits, el número de elementos que formen la red es mayor (2^{64}) que en el de 16, pero las direcciones de cada módulo no pueden ser definidas arbitrariamente por el usuario como ocurre con el direccionamiento de 16 bits y hay que utilizar el número serial que viene de fábrica.

El número máximo de direcciones de red que se pueden asignar a los dispositivos es de $2^{16}=65535$, que coincide con el número máximo de elementos que pueden formar una red ZigBee cuando se utilizan direcciones de 16 bits.

En la red diseñada para este proyecto se van a direccionar los módulos con 16 bits ya que no se necesita un número elevado de elementos.

La dirección de origen de 16 bits del módulo se define arbitrariamente mediante el comando MY (con un rango entre 0x0 y 0xFFFE) y la dirección de destino se define mediante el comando DL y la utilización del *software* XCTU.

4.3. Tipos de Redes que soporta XBee.

Los módulos XBee, pueden ser ajustados para usarse en redes de configuración punto-a-punto o punto-a-multipunto. Teniendo en cuenta que los módulos XBee Serie 2 permiten hacer redes *mesh*, y la serie 1 no, que es la de la que se dispone en este proyecto.

- **Conexión Punto a Punto.** Un módulo se conecta con otro como se muestra en la Figura 10. Para ello se utilizan los comandos MY y el DL para configurar la dirección. Este tipo de conexión ha sido el utilizado durante el desarrollo de la primera parte de este proyecto hasta conseguir una familiarización con los dispositivos XBee y su funcionamiento.



Figura 10. Conexión Punto a Punto.

- **Conexión Punto a Multipunto.** Permite transmitir información desde la entrada serie de un módulo a uno o varios módulos conectados a la misma red de manera más controlada, ya que se necesitan las direcciones de los otros módulos, por lo que existe mayor seguridad. Para esto se necesitan dos comandos más aparte de MY y DL. Se utilizará el direccionamiento de 16 bits.

El primer comando es el ID de la PAN (*Personal Area Network*- Red de Área Personal). Todos los módulos que tengan idéntico PAN ID, pertenecerán a la misma red. El comando para configurar este valor es ID, es decir, ATID, y su rango va entre 0x0 y 0xFFFF. En la Figura 11 está representada este tipo de red punto a multipunto y como se configura.

La elección del canal debe ser cuidadosa, ya que otras tecnologías como WiFi o bluetooth utilizan el mismo espectro de frecuencias, por lo que se podría producir interferencia.

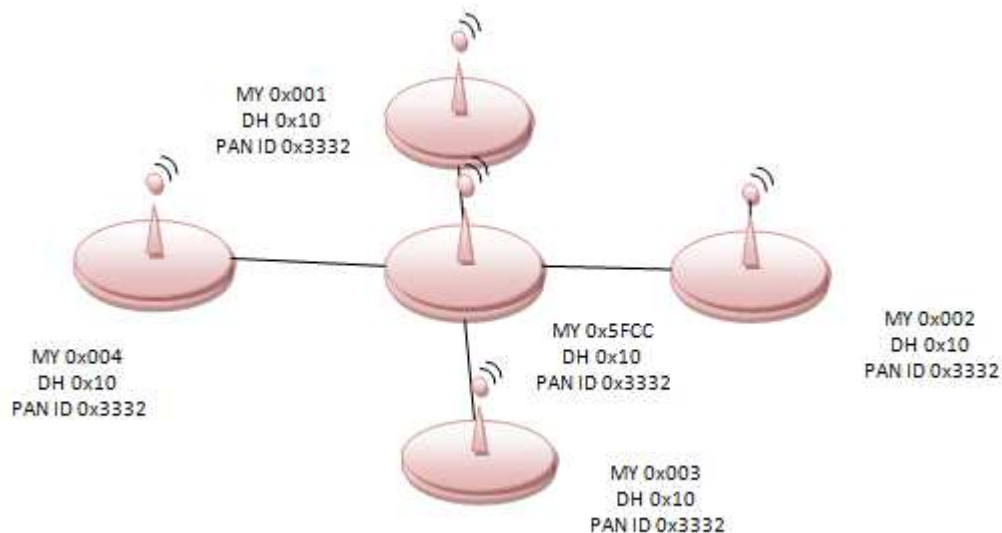


Figura 11. Conexión Punto a Multipunto.

- **Conexión NonBeacon. Peer to Peer.** Una red *peer-to-peer* permite que todos los módulos se conecten con todos, es decir, se crea una conexión de par en par con cada uno de los módulos de la red. El modo de conexión *Non-Beacon* es la configuración por defecto y permite establecer una red *peer-to-peer* donde cada módulo puede hacer las funciones de maestro o esclavo.

La configuración de red *Non-Beacon*, se refiere a que cada nodo primario (o nodo central maestro) se mantiene despierto siempre. Por lo que los demás dispositivos que se conectan a él, pueden entrar en modo *SLEEP* (ahorro de energía), y sólo despertarse cuando sea necesario para enviar datos. En una red *Beacon*, los dispositivos enrutadores (o nodos primarios) están siempre en modo *SLEEP*, y envían señales de su existencia (llamadas *Beacon*) cada cierto intervalo al resto de la red. Así para poder comunicarse, deben estar todos los dispositivos totalmente organizados, ya que de no ser así, existe la posibilidad de perder la señal *Beacon* y no poder enviar hasta la próxima entrega. La ventaja de las redes *Beacon*, es el ahorro de energía. Por este motivo las redes *Non-Beacon* están pensadas para

dispositivos que posean una alimentación segura, mientras las *Beacon*, para alimentación autónoma, como baterías. Los módulos XBee Series 1, sólo soportan redes *Non-Beacon*.

- **Conexión *Non-Beacon* con coordinador.** Es básicamente lo mismo que una red punto-multipunto con la diferencia de que existe un módulo central que posee ciertas propiedades y características que le permiten administrar la red. En esta red, el módulo central es llamado coordinador, mientras que el resto de módulos son llamados Dispositivos Terminales (*End Device*) como se representa en la Figura 12. Un mismo módulo XBee puede ser configurado para funcionar como coordinador o como Dispositivo Terminal.

Para configurar esta red, todos los módulos deben tener el mismo canal y la misma PAN. El módulo coordinador se configura como ATCE=1 (ATCE1), mientras que todos los demás, los cuales serán llamados Dispositivos Terminales, se configuran como ATCE=0 (ATCE0).

Este tipo de configuración se usa cuando se necesita una unidad central para enviar mensajes a varios módulos, o juntar información proveniente de varios Dispositivos Terminales, asignar canales o ID de redes PAN.

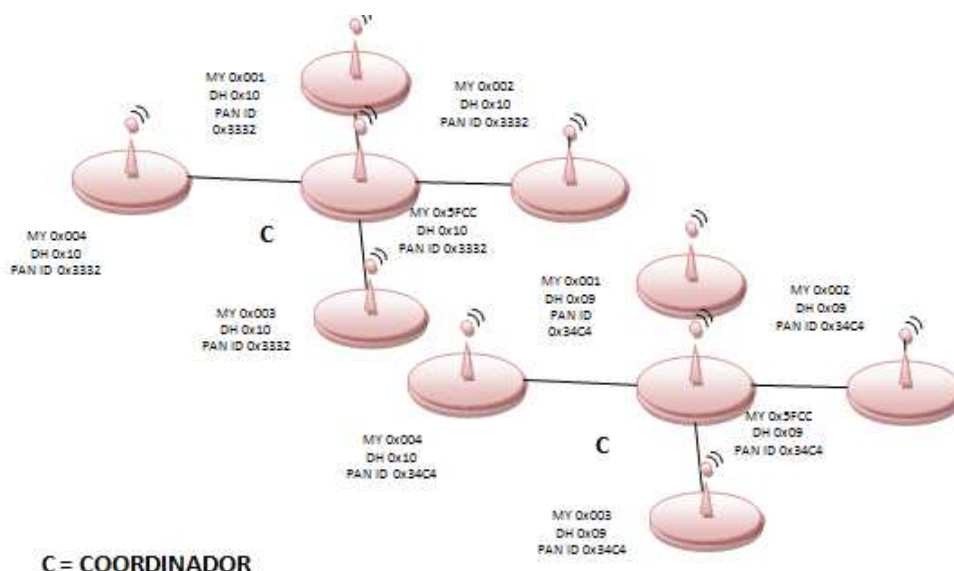


Figura 12. Red *NonBeacon* con coordinador utilizada en este proyecto.



En este proyecto se ha optado por una conexión *Non-Beacon* con coordinador. Los dispositivos terminales de la red serán los módulos Arduino-XBee emisores, y el coordinador será el módulo Arduino-XBee receptor que se encuentra unido al PC y es el encargado de procesar toda la información que le llegue de los distintos módulos.

Los XBee con los que se cuenta son de Serie 1, por lo que no se podría realizar una red *Beacon*, y como no se necesita comunicación entre los diferentes *End Points*, no se ha optado por una red punto a multipunto.



Capítulo 5. Descripción del sistema.

En este capítulo se da una visión general del diseño del sistema implementado en este proyecto. Se incluye una descripción de cada elemento de sistema y una breve explicación del funcionamiento de cada bloque junto con la configuración de los elementos que forman dicho bloque.

La Figura 13, muestra los diferentes bloques que forman la WSN y las conexiones entre ellos, existiendo una conexión inalámbrica bidireccional entre cada nodo y el coordinador y una conexión física cableada entre cada nodo con sus correspondientes sensores y entre el coordinador y el PC.

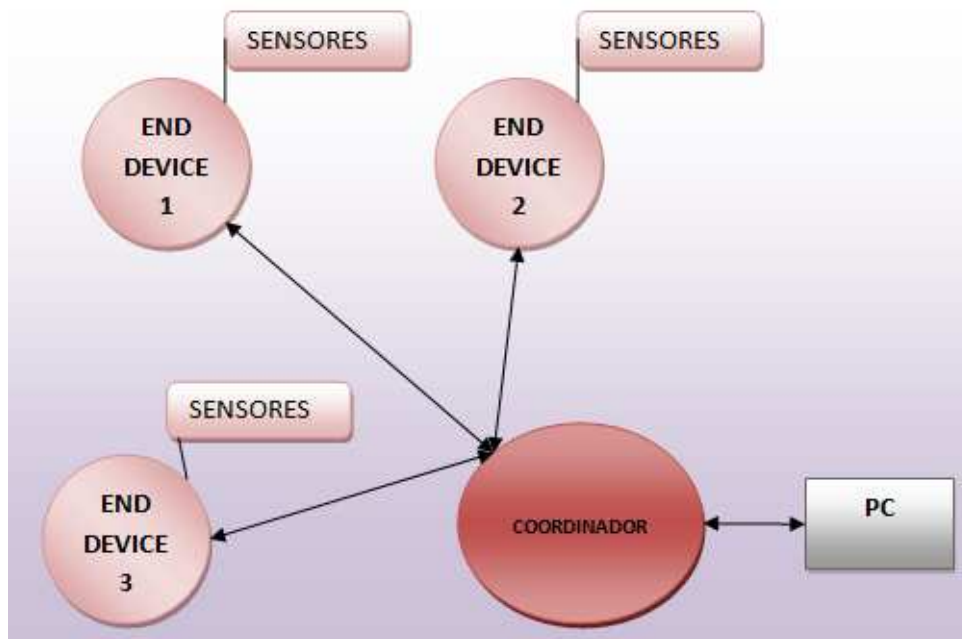


Figura 13. Esquema del sistema implementado.

El sistema está dividido en cuatro bloques:

- Bloque Emisor. Con tres dispositivos *end device* encargados de transmitir y recibir datos. Cada dispositivo cuenta con un Kit Arduino Diecimila, un módulo de comunicación por RF XBee y un cable USB A-B o un cargador de red.
- Bloque Receptor, que cuenta con un dispositivo coordinador (un kit Arduino Diecimila) que recibe los datos del bloque emisor y los gestiona con el PC. También manda información a cualquiera de los nodos mediante el módulo de comunicación XBee que posee.
- Bloque de Sensado formado por cuatro elementos sensores fabricados con fibra óptica de plástico (POF) que recogen información del medio, con sus respectivos emisores y fotodiodos.
- Bloque de Procesado de datos. El PC al que va unido el equipo coordinador se encarga de monitorizar los datos mediante un entorno gráfico que será diseñado mediante LabWindows.

5.1. Elementos del Sistema.

5.1.1. Coordinador y Emisores.

El elemento coordinador y los elementos *end devices* se dividen en dos bloques: el bloque receptor y el bloque emisor. Como se puede ver en la Figura 14, el bloque receptor se corresponde con el elemento coordinador y está formado por una placa Arduino y un módulo XBee, mientras que el bloque emisor se corresponde con los tres dispositivos *end devices* formados por una placa Arduino y un módulo XBee cada uno.

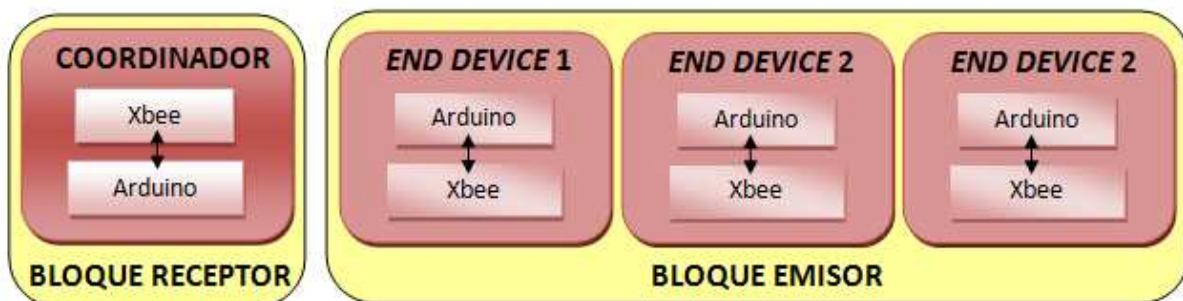


Figura 14. Diagrama de bloques del bloque receptor y del bloque emisor.

Tanto el elemento coordinador como los tres *end devices* están formados por un Kit Arduino Diecimila como el que aparece en la Figura 15. Un Arduino es una plataforma *open-hardware* basada en una sencilla placa con entradas y salidas (I/O), analógicas y digitales, y en un entorno de desarrollo que implementa el lenguaje *Processing/Wiring*. Funciona en Windows, Linux, Mac OS X y GNU/Linux (Debian). [Arduino]

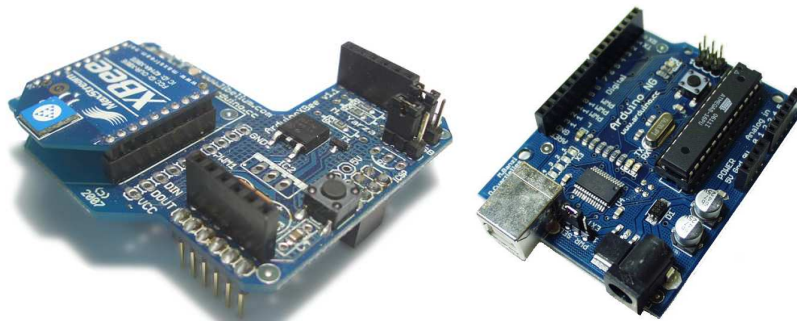


Figura 15. Placas Arduino Diecimila utilizadas. [Arduino]

Al ser *open-hardware* tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para desarrollar cualquier tipo de proyecto sin tener que adquirir ningún tipo de licencia.

La placa USB es una evolución de la placa serie que incorpora un puerto USB para comunicarse con el PC.

Viendo sus componentes y comenzando en el sentido de las agujas del reloj desde el centro de la parte superior de la Figura 16 que muestra el esquemático de Arduino se tiene: **[Arduino]**

- Pin de referencia analógica (naranja)
- Señal de tierra digital (verde claro)
- Pines digitales 3-13 (verde)
- Botón de reset (azul oscuro)
- Pines de entrada analógica 0-5 (azul claro)
- Pines de alimentación y tierra (alimentación: naranja, tierra: naranja claro)
- Entrada de la fuente de alimentación externa (9-12V DC) – X1 (rosa)
- Puerto USB (amarillo).

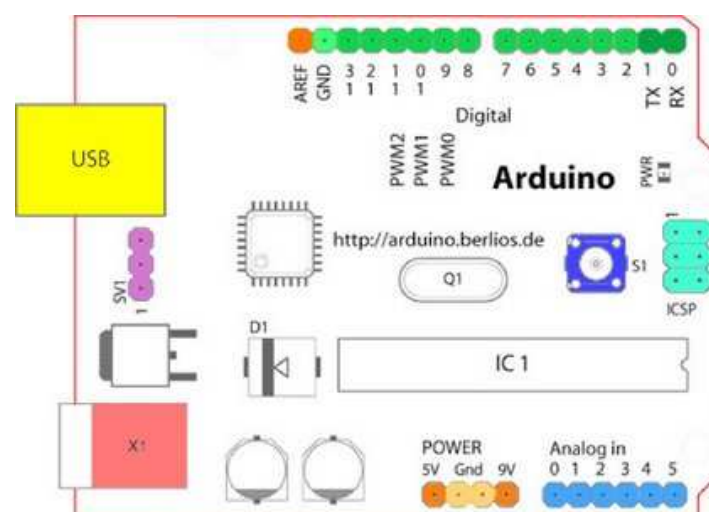


Figura 16. Esquemático Placa Arduino.[Arduino]



Su chip principal es el Atmega8 y 168, un chip sencillo y de bajo coste que permite el desarrollo de múltiples diseños. Algunas de las características del Arduino son:

- Voltaje Operativo: 5V
- Voltaje de entrada: 6-12 V
- Entradas/Salidas Digitales: 14 Pines (De los cuales 6 cuentan con salida PWM)
- Entradas Analógicas: 6 Pines
- DC por cada E/S: 40 mA
- Memoria Flash: 16 KB (2 KB son utilizados *bootloader*)
- SRAM: 1 KB
- EEPROM: 512 bytes
- *Clock Speed*: 16 MHz

El bloque coordinador cuenta con un Arduino Diecimila, un módulo XBee y un cable USB A-B que le une al PC, encargado de recoger e interpretar los datos recibidos por cada *end device* y transmitirlos al bloque del PC para que éste se encargue de mostrarlos por pantalla al usuario final.

Cada bloque *end device*, nodo emisor, está formado por un Arduino Diecimila, que cuenta con un módulo XBee, y un cable USB A-B o un cargadora la red. Cada nodo tiene la función de detectar las variaciones que se produzcan en los sensores que a él van unidos y transmitirle los nuevos valores de medida cuando se produzca una modificación en el equipo sensor al coordinador.

De los tres nodos emisores que tiene el sistema, uno de ellos, el llamado equipo tres, se va a utilizar para hacer pruebas sobre la respuesta de transmisión de los módulos XBee en relación con la distancia del equipo de medida y las paredes que puede encontrarse de por medio la transmisión. Debido a la necesidad de desplazamiento del nodo tres, las entradas sensoras se simularán con una fuente de alimentación.

Los otros dos nodos sensores, el equipo uno y el equipo dos, se van a utilizar para sensar el nivel de líquido de un depósito y la variación de temperatura del habitáculo. Cada nodo tiene unido a él dos POFs, que se explicarán en el siguiente punto, un nodo se encarga de medir el nivel de líquido y el otro medirá las variaciones de temperatura.

Los tres nodos recogen los datos correspondientes a cada equipo y se los envían por RF al coordinador cada vez que se produzca una modificación en la medida recogida. A su vez, el equipo coordinador puede requerir que los nodos sensores hagan una nueva medida cuando el usuario quiera, desde la aplicación gráfica, para verificar que los datos recogidos en cada instante son los reales.

5.1.2. Sensores.

Como se ha mencionado anteriormente, cada nodo emisor cuenta con dos sensores: al nodo uno y al nodo dos van unidos dos sensores de POF, mientras que el nodo tres se utiliza para comprobar el alcance de los nodos emisores. En total, el sistema cuenta con cuatro elementos sensores colocados dos a dos. Dos van unidos al equipo uno, midiendo temperatura, otros dos al equipo dos midiendo niveles de líquido discretos y las dos entradas de medida de alcance como se puede ver en el diagrama de bloques emisor que se muestra en la Figura 17.

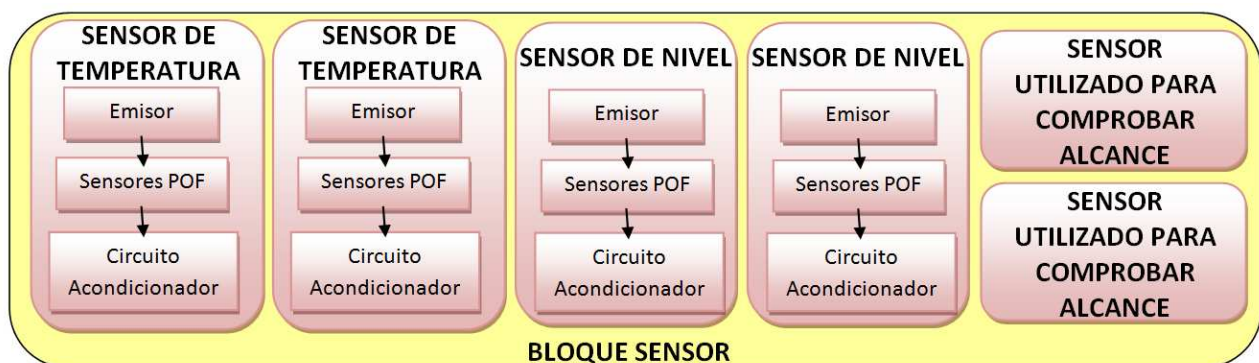


Figura 17. Diagrama de bloques del bloque sensor.

La elección de fibra óptica de plástico se ha basado en la ventaja de que por ella no circula corriente eléctrica, lo que hace accesible su uso en ambientes peligrosos y de difícil acceso. La fibra óptica es un conductor óptico capaz de soportar la transmisión de haces de luz gracias a continuas reflexiones ocurridas en su interior debido a los materiales de los que está compuesta.

El nivel de potencia que posee la señal luminosa a la entrada de la fibra es mayor que el nivel que posee al llegar al final de ésta, lo que quiere decir que la señal se ha atenuado.

Una señal se atenúa por varios factores que dependen de las propiedades físicas de la fibra y su comportamiento al transmitir un haz de luz por ella. Cuando la fibra óptica es sometida a una curvatura, se genera una atenuación y estas pérdidas pasan desapercibidas hasta que se sobrepasa una curvatura crítica donde los haces de luz logran escapar del núcleo por superar el ángulo máximo de reflexión total interna.

En un sistema de transmisión por fibra óptica existe un emisor encargado de convertir señales eléctricas en ópticas, que serán transmitidas a través de la fibra. Una vez transmitida la señal luminosa, en el otro extremo del circuito se encuentra un tercer componente llamado detector óptico que convierte las señales ópticas en eléctricas. En la Figura 18 se observan los bloques principales del sistema de transmisión de fibra óptica formados por: un LED emisor, un cable de fibra óptica y un fotodiodo unido a una resistencia para transformar la intensidad que por él circula en tensión.

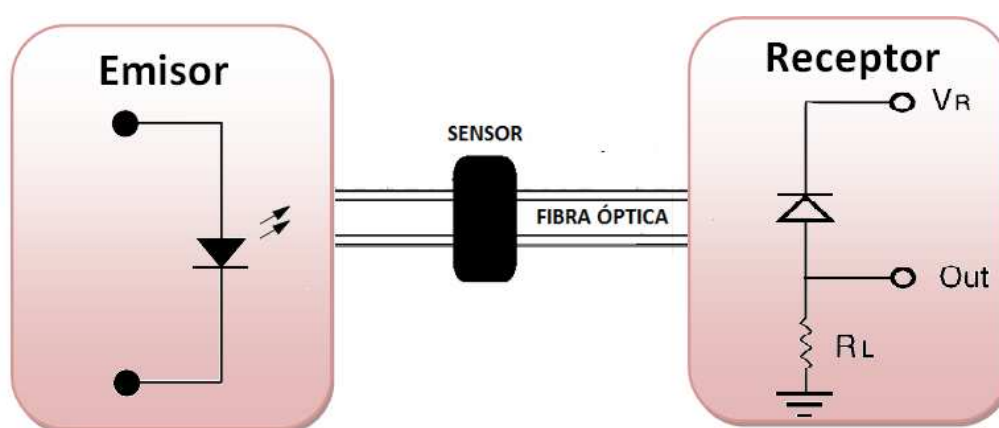


Figura 18. Sistema de transmisión por fibra óptica.

El sistema de emisión encargado de convertir señales eléctricas en ópticas utilizado para este proyecto es un LED integrado en una placa de emisión de Hewlett Packard (HFBR - 1528), cuyo esquemático puede verse en la Figura 19 , que emite con una longitud de onda de 650 nm, con una potencia máxima de emisión de 1mW, conexión para POF, y driver incorporado.

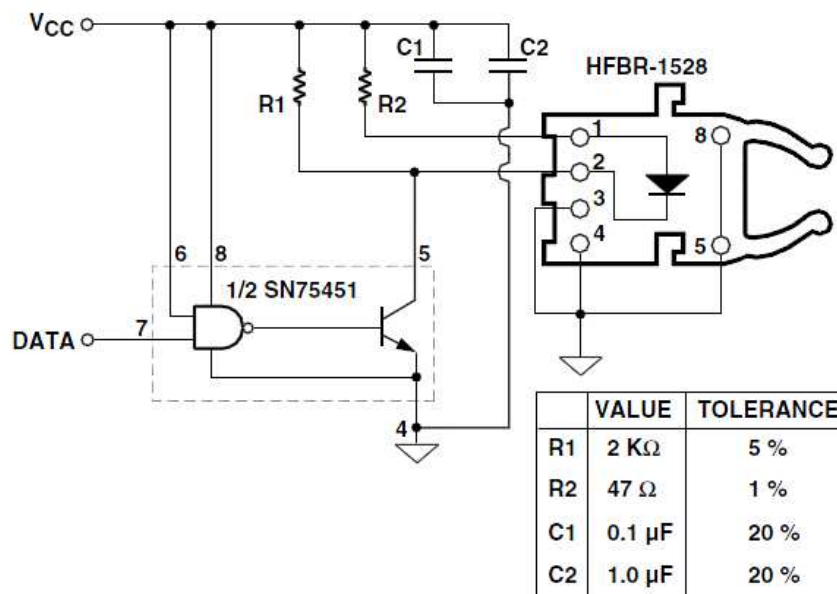


Figura 19. Emisor HFBR – 1528. [Datasheet]

El sistema de recepción que convierte las señales ópticas en eléctricas mostrado en la Figura 20, cuenta con un fotodiodo IF-D91 que convierte la luz procedente de la fibra óptica en corriente, posee un encapsulado para fibra óptica y una responsividad de 0,2 A/W para una longitud de onda $\lambda=660\text{nm}$, y una resistencia que transforma la corriente suministrada por el fotodetector en tensión.

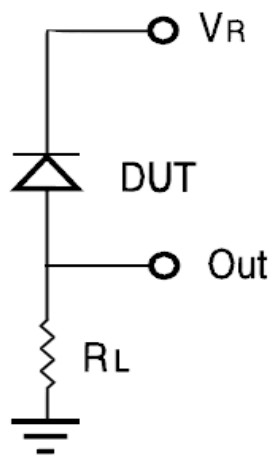


Figura 20. IF D91. [Datasheet]

En el Grupo de Displays y Aplicaciones Fotónicas de la Universidad Carlos III de Madrid se han llevado a cabo diferentes trabajos relacionados con la fibra óptica de plástico como sensor de nivel y de temperatura. A continuación se describe brevemente el principio de medida seguido.

En el caso del sensor de nivel, la variación del índice de refracción del medio externo provoca que varíen las condiciones de propagación en la fibra óptica, al variar las condiciones de reflexión total. Así en el caso de los sensores de nivel, se produce una variación del índice de refracción del aire con $n=1$, a agua con $n= 1.33$ y a aceite con $n=1.46$.

Una descripción de su influencia, se puede encontrar en [**Mon+10**] donde se describe un sensor de nivel basado en una zona sin recubrimiento y con macro-curvaturas para aumentar la sensibilidad, o en [**Mon+09**] donde se produce un pelado y además se acopla la luz a otra fija para conseguir una medida auto-referenciada insensible a perturbaciones de la potencia óptica ajenas al principio de medida.

En el caso de los sensores de temperatura (T), al variar esta magnitud se produce una variación del índice de refracción del medio de la cubierta que rodea al núcleo ($n_{cladding}$) según la ecuación:

$$n_{cladding} = n_{cladding0} + \alpha_n \cdot (T - T_0) \quad (1)$$

Donde $n_{\text{cladding } 0}$ es el índice de refracción de la cubierta a la temperatura de referencia (T_0) y α_n es el coeficiente termo-óptico del material que forma la cubierta. Para la fibra POF utilizada en los ensayos, este coeficiente termoóptico es igual a $-5 \cdot 10^{-4} \text{ } ^\circ\text{C}^{-1}$ y el índice de refracción de la cubierta es igual a 1,417 a $T_0 = +25 \text{ } ^\circ\text{C}$, esto se puede encontrar de una forma más detallada en [Tap+11].

Una vez analizado el principio de funcionamiento de los sensores, se ha optado por utilizar en este proyecto dos sensores de nivel y dos de temperatura de POF.

A las cuatro fibras utilizadas como sensores se les ha eliminado el revestimiento mecánico, realizando en dos de ellas también un pulido de su cubierta, para obtener dos medidas de temperatura y dos de nivel de una forma más sensible. La Figura 21 cuenta con las dos imágenes de los sensores de fibra óptica implementados, la de la izquierda se corresponde con el sensor de temperatura, formado por una fibra pulida y otra sin pulir y una placa calefactora, y la de la derecha con el sensor de nivel formado por una fibra pulida y otra sin pulir y dos depósitos donde medir el nivel de líquido.

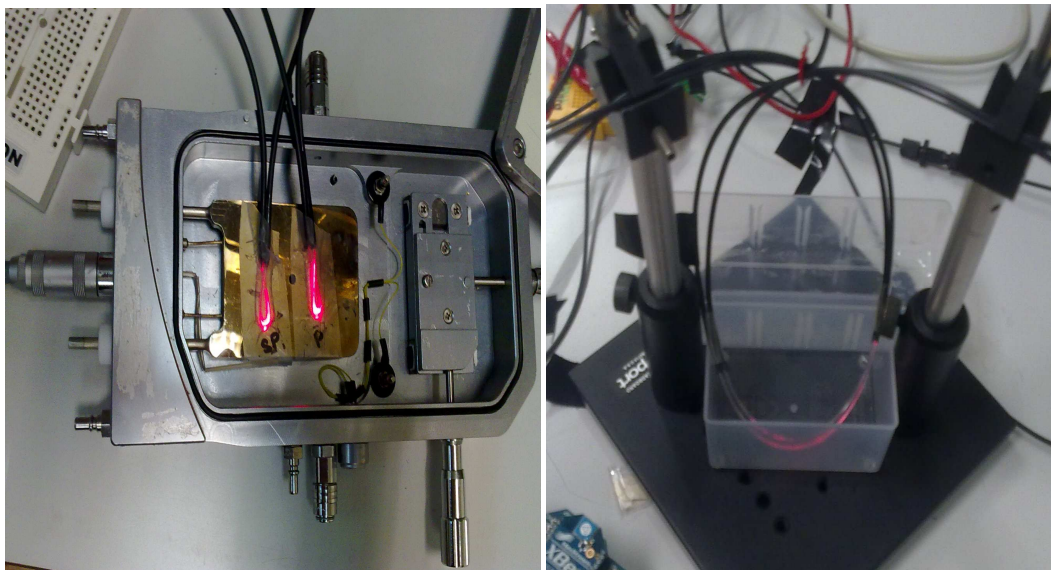


Figura 21. Sensor de Temperatura y Sensor de Nivel implementados en cada nodo emisor.

Cada fibra dispone de un sistema de emisión (HFBR-1528) como los que aparecen en la Figura 22, encargado de generar la luz que transporta la fibra óptica al depósito y a la placa calefactora, y que

posteriormente será recibida por el fotodiodo que convierte la luz procedente de la fibra óptica en corriente.

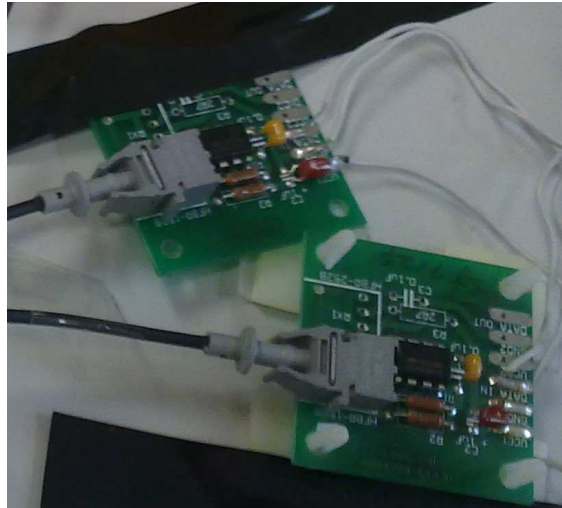


Figura 22. Emisor HFBR-1528 utilizado.

5.1.3. Procesamiento de datos.

Los datos sensados por cada POF son enviados inalámbricamente de cada nodo emisor al nodo coordinador. Éste se encuentra conectado mediante USB a un PC y se encarga de transmitirle la trama recibida. Dichos datos son procesados y monitorizados en una interfaz gráfica diseñada en LabWindows.

La aplicación monitoriza los valores de tensión de cada elemento sensor, y las variaciones que se producen en cada medida. Dando la posibilidad al usuario de testear todos los nodos para comprobar las medidas en cualquier momento. El programa también cuenta con una pequeña alarma que salta cada vez que se sobrepasa un valor, avisando de que algo ocurre en ese punto porque la fibra se ha saturado.



5.2. Funcionamiento del sistema.

Cada módulo emisor se encuentra unido a dos sensores cuya respuesta variará entre 0 y 5 Voltios dependiendo de las pérdidas de potencia óptica que se produzcan en la fibra óptica al sumergirla en los líquidos de los depósitos o al variar la temperatura de la placa calefactora. Cuando se produzca una variación en la entrada de datos analógica correspondiente a alguno de los dos sensores, el microprocesador integrado en la placa Arduino realiza una lectura de los datos de entrada, hace una conversión analógica-digital, y le envía una serie de datos al módulo coordinador: dirección del dispositivo, canal de lectura, si el valor de medida sobrepasa el límite establecido de 5 voltios y la lectura de datos de la entrada analógica para que sean mostrados en la pantalla del PC.

El microprocesador del Arduino emisor le envía mediante una comunicación serie al módulo XBee los datos que se desean transmitir al coordinador por radiofrecuencia, y éste cuando los recibe, se los pasa al PC por el puerto USB, para que sean monitorizados por pantalla.

Una vez recibidos los datos en el PC, éstos son mostrados por pantalla mediante una interfaz gráfica que se ha creado para que resulte más sencillo de visualizar el resultado.

A continuación se muestra el esquemático correspondiente a las conexiones entre el emisor HFBR 1528, la fibra óptica, el receptor IF D91 y la placa Arduino perteneciente al nodo emisor. En la Figura 23 aparece dicha conexión, mientras que en la Figura 24 se representa el esquemático de la placa Arduino Diecimila completo.

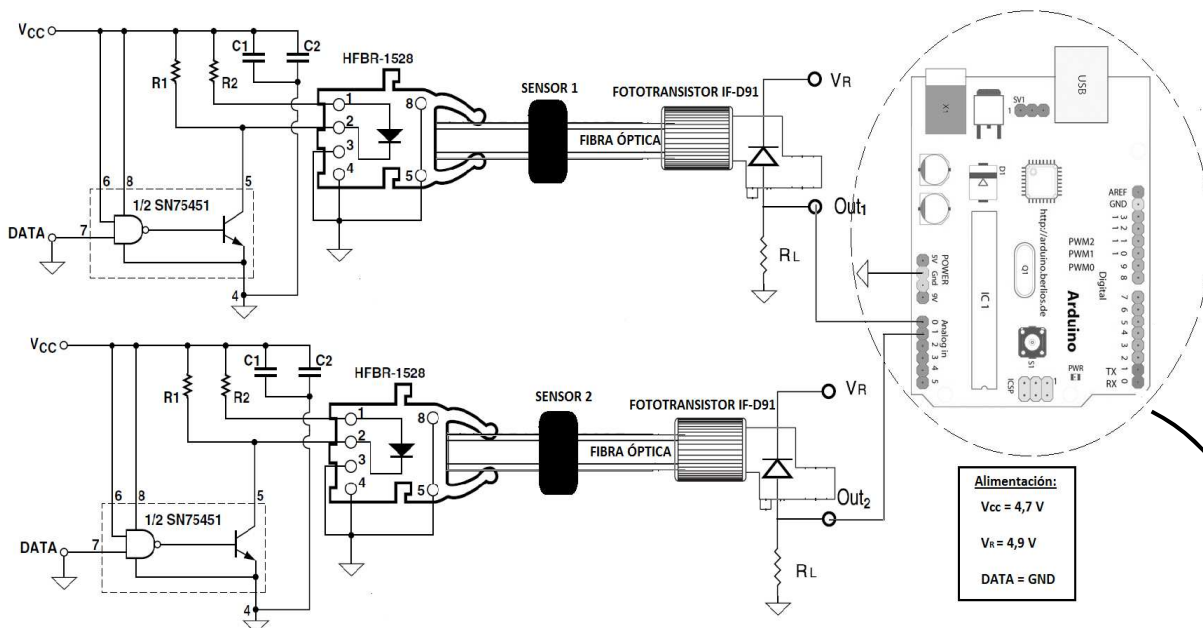


Figura 23. Esquemático Emisor – POF – Fotodiodo – Arduino.

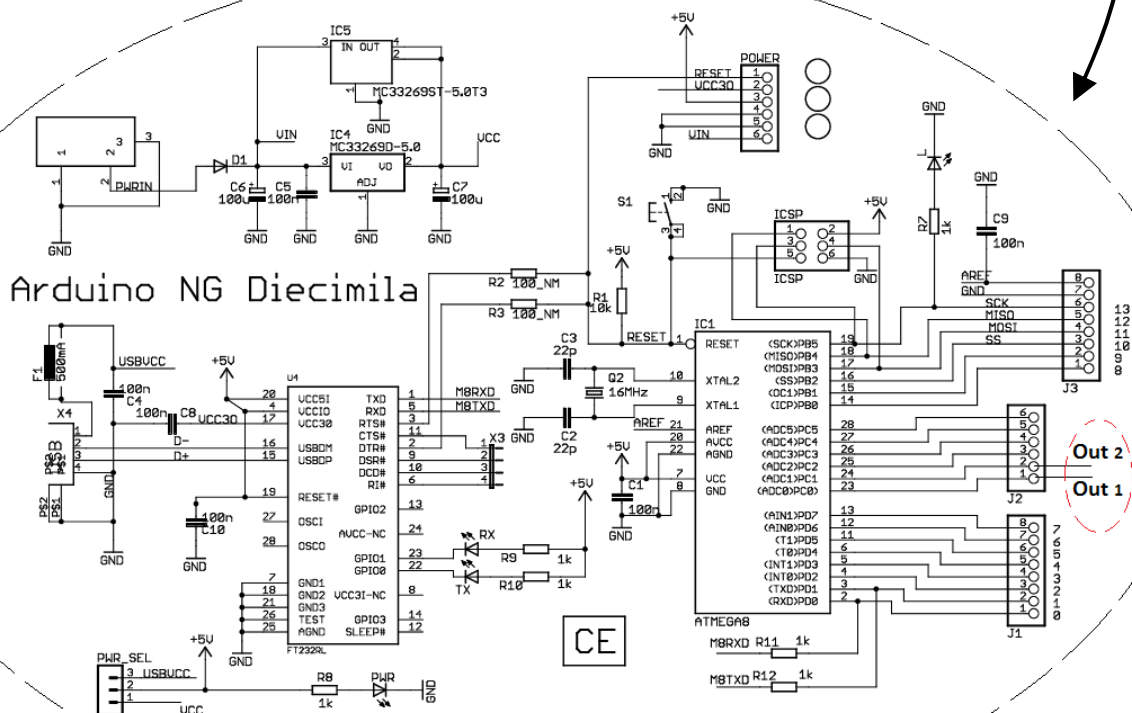


Figura 24. Esquemático Placa Arduino.[Arduino]

5.3. Configuración del Sistema.

En este proyecto se ha optado por una conexión punto a multipunto como se ha explicado anteriormente. Los dispositivos terminales de la red serán los módulos Arduino-XBee emisores, y el coordinador será el módulo Arduino-XBee receptor que se encuentra unido al PC y es el encargado de procesar toda la información que le llegue de los distintos módulos.

Para configurar una red punto a multipunto lo primero que debe hacerse es configurar cada módulo XBee con su respectiva etiqueta, como coordinador o como *end device*. Para ello se utiliza el programa X-CTU, véase anexo 2 para conocer su funcionamiento, y dependiendo de la misión de cada dispositivo los valores de los parámetros de configuración serán diferentes como se puede ver en la Tabla 3.

	COORDINADOR	END DEVICE 1	END DEVICE 2	END DEVICE 3
Destination Address(DH)(DL)	0	0	0	0
Source Address(MY)	0	1	2	3
Coordinator Enable(CE)	1	0	0	0
Api Enable(AP)	1	1	1	1

Tabla 3. Parámetros de configuración de los módulos XBee.

Una vez configurados los módulos XBee, el siguiente paso es realizar el código fuente usando el *software* de Arduino, véase anexo 3 donde se explica el modo de utilización de dicho programa, y cargarlo al microprocesador de cada kit Arduino mediante USB.

El siguiente paso es crear la interfaz gráfica donde se analizarán todos los datos recogidos por el sistema. Esta interfaz gráfica se realiza con el *software* LabWindows, en el anexo 1 se explica el modo de utilización de dicho programa, que nos permite monitorizar los resultados de medida de una forma más sencilla y rápida.



Capítulo 6. Pruebas realizadas/ Fases de desarrollo.

Según los objetivos de este proyecto se quiere realizar un control sensorial por radiofrecuencia, por lo que se han ido realizando pruebas independientes divididas en distintas fases para después unir todo.

El diseño y la implementación de la WSN se ha dividido en tres fases de trabajo, donde en cada una de ellas se han realizado varios ensayos.

En este capítulo se abarca cada una de las fases del proyecto, detallando los componentes del sistema, las características de cada red y la interfaz gráfica diseñada para cada parte desarrollada.

La primera fase se basa en una toma de contacto con los módulos Arduino y con la comunicación por RF, cuyo fin es obtener una red punto a punto (formada por un nodo sensor y un receptor) capaz de leer seis sensores conectados a un mismo emisor y monitorizar los datos. Por ello se ha dividido en tres ensayos: control módulo Arduino, comunicación por RF entre módulos y monitorización de datos.

En la segunda fase se pasa de una red punto a punto, diseñada anteriormente, a una red punto-multipunto formada por tres nodos sensores (con dos sensores por nodo) y un coordinador. La comunicación entre los nodos sensores y el coordinador es unidireccional, cada nodo envía tramas de datos al coordinador y este no puede enviarle tramas de datos a los dispositivos finales.

Por último, en la tercera fase, se llega al diseño final de la WSN punto-multipunto. Esta red es la misma que la de la fase dos, con la diferencia de que aquí se consigue una comunicación bidireccional entre los nodos sensores y el coordinador permitiendo el envío de tramas de datos del coordinador a los nodos sensores.

A continuación, la Tabla 4 contiene un breve resumen con las características de la red más importantes a destacar de las diferentes fases de desarrollo.

	FASE I	FASE II	FASE III
TIPO DE RED	Punto a punto	Punto a Multipunto	Punto a Multipunto
COMUNICACIÓN	Unidireccional	Unireccional	Bidireccional
Nº ARDUINOS UTILIZADOS	2	4	4
MODO FUNCIONAMIENTO XBee	Transparente	API	API

Tabla 4. Resumen de las características de cada Fase de desarrollo.

6.1. Primera fase de desarrollo del proyecto.

En esta primera fase se pretende obtener una red punto a punto como la que aparece en la Figura 25, capaz de monitorizar los datos recogidos por seis sensores que se encuentran unidos a un mismo nodo emisor por RF.

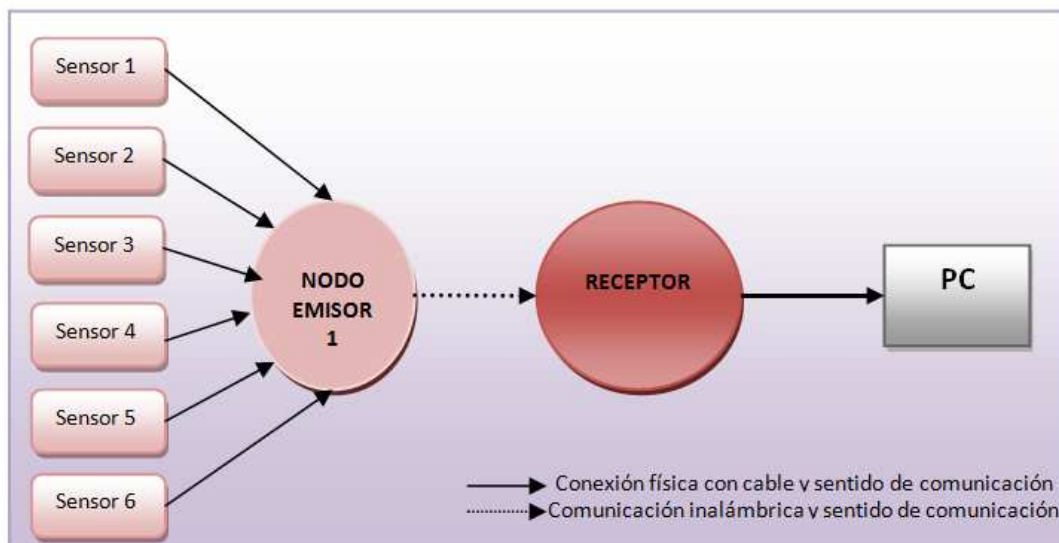


Figura 25. Sistema diseñado en la Fase I.

Se ha dividido la Fase I en tres pruebas:

- Control del módulo emisor y comunicación de éste con dos sensores.
- Implementación de una red punto a punto con comunicación por RF entre el módulo emisor y el coordinador, y monitorización de los resultados leídos por los dos sensores en una interfaz gráfica.
- Integración de cuatro sensores más a la red, monitorizando los datos recogidos por los seis sensores que forman la red punto a punto en esta tercera prueba.

La primera prueba ha consistido en conseguir una comunicación del módulo emisor con los sensores, es decir, se realiza una lectura analógica de las distintas entradas, y esta lectura se realiza cada vez que se produzca una variación en la entrada analógica o una interrupción externa.

Una vez se consigue este punto, el siguiente paso es comunicar por radiofrecuencia los módulos emisor y receptor para sacar por la pantalla del PC los datos obtenidos.

En el segundo ensayo el módulo receptor no dispone de microprocesador, por lo que simplemente le pasan los datos por comunicación serie al PC y éste los muestra por pantalla, sin la posibilidad de gestionar los datos recibidos o recibir datos de distintos módulos. Aquí aún no ha sido diseñada la interfaz gráfica, y los datos se muestran por el *hyperterminal*.

En la tercera prueba, una vez conseguida la comunicación entre módulos, se le incorpora al kit receptor su microprocesador, por lo que ahora el emisor solo transmite los valores de las lecturas analógicas para que sea el receptor el encargado de tramitar todos los datos y así poder recibir datos de distintos sensores.

Tras comunicar los módulos emisor-receptor satisfactoriamente, se ha obtenido una red punto a punto capaz de realizar una lectura analógica de cada una de las 6 entradas disponibles en la placa Arduino emisora, transmitir dichos datos cada vez que se produce una variación en la medida, y procesar esos datos en el dispositivo receptor.

Para facilitar la lectura de los datos finales, se propone la realización de una interfaz gráfica que aporte rapidez y sencillez a la hora de interpretar las medidas recibidas. Este primer diseño de la aplicación puede verse en la Figura 26, donde sólo se cuenta con un reloj analógico para representar los valores de sólo dos sensores y un display donde leer las medidas.



Figura 26. Primera interfaz gráfica diseñada en la Fase I.

Seguidamente, se busca hacer más intuitiva la pantalla del sistema de medida, donde el usuario, a simple vista pueda apreciar las variaciones de una medida, y para ello se mejora la aplicación una vez se ha comprobado que la red punto a punto diseñada es capaz de incorporar seis sensores de medida en un solo dispositivo emisor. El resultado final de dicho diseño puede verse en la Figura 27.

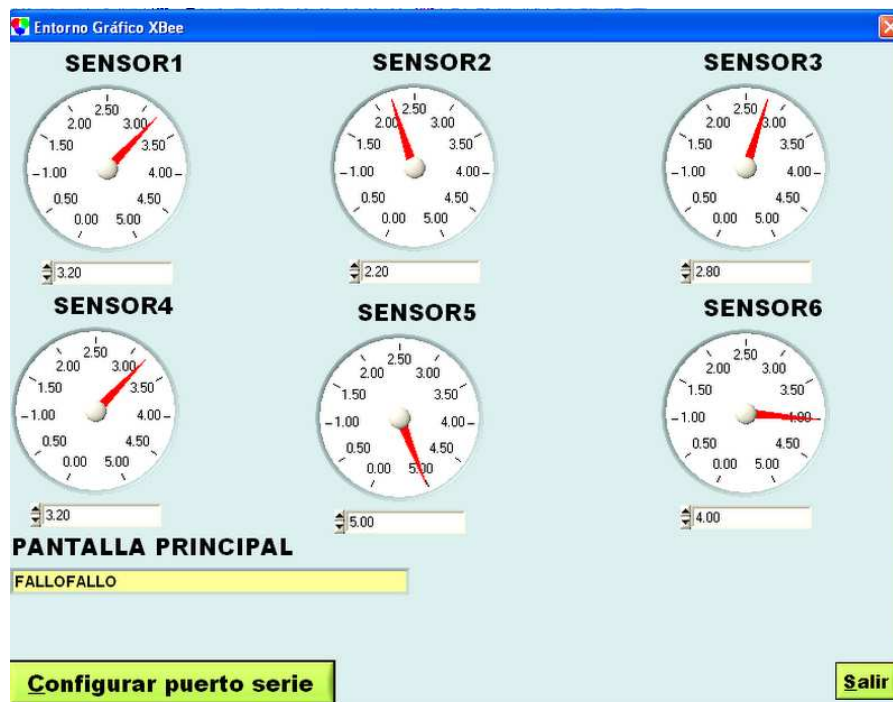


Figura 27. Segunda interfaz gráfica diseñada en la Fase I.

La pantalla donde aparecen representados las diferentes medidas de los seis sensores que es capaz de controlar el Arduino – Emisor cuenta con:

- Seis relojes de medida, pertenecientes al único equipo emisor del sistema, y cada reloj representa cada uno de los sensores unidos al emisor.
- Una pantalla donde cada vez que algún sensor sobrepasa los 5 Voltios establecidos como límite máximo muestra la palabra FALLO para que el usuario interprete que algo va mal. Se puede decir que hace de alarma, pero no informa sobre el equipo en el que se da la alarma, eso se intentará mejorar en los siguientes puntos.

- Los botones de configuración del puerto serie de comunicación y el botón de salir de la aplicación.

La Tabla 5 recoge las principales características de la red punto a punto llevada a cabo en esta primera Fase de trabajo.

CARACTERÍSTICAS PRINCIPALES DE LA FASE I	
	FASE I
Tipo de red	Punto a Punto
Tipo de comunicación	Unidireccional
Nº kits Arduino en la red	2
Nº <i>end devices</i>	1
Nº <i>end devices</i> máx. que soporta la red	1
Nº sensores por <i>end device</i>	6
Control de envío y recepción de datos	NO
Modo programación módulos	Transparente
Microprocesador coordinador	SÍ
Disposición JUMPERS de comunicación en coordinador	XBee

Tabla 5. Características de la red punto a punto desarrollada en la Fase I.

6.2. Segunda fase de desarrollo del proyecto.

Se ha tenido como objetivo en esta segunda Fase el empezar a diseñar la red sensorial con la que se quiere trabajar al finalizar este proyecto, una red formada por un coordinador y tres dispositivos finales con dos sensores por nodo. La red punto a multipunto desarrollada en esta parte se puede ver en la Figura 28.

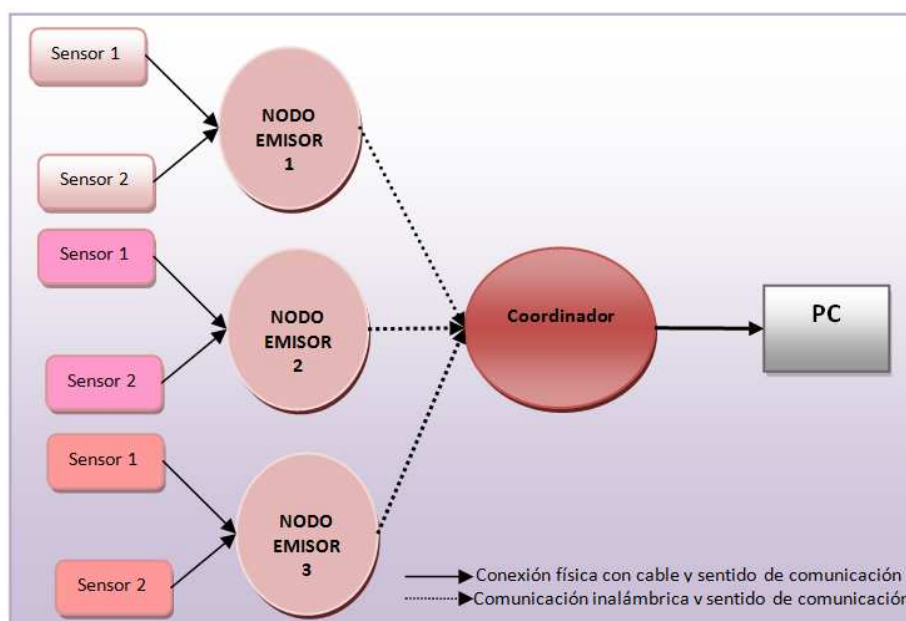


Figura 28. Sistema diseñado en la Fase II.

Hasta ahora el sistema estaba formado por dos módulos Arduino-XBee (un receptor y un emisor) programados en modo Transparente, que viene por defecto en los módulos XBee, donde cada módulo es programado con la dirección de origen y destino con la que van a trabajar.

El modo Transparente es el idóneo para trabajar en comunicaciones punto a punto ya que no es necesario ningún control en el envío y recepción de datos.

Como se va a proceder a la introducción de más módulos en la red, se van a programar los módulos en modo API, anteriormente operaban en modo transparente, donde toda la información enviada y recibida forma parte de una trama de datos (*framing*), con una estructura definida que permite una forma más robusta para enviar datos.

Cuando la configuración API está activada, cada paquete RF que se envía o recibe se encapsula en un *frame* de datos UART. El *frame* se observa en la Figura 29:

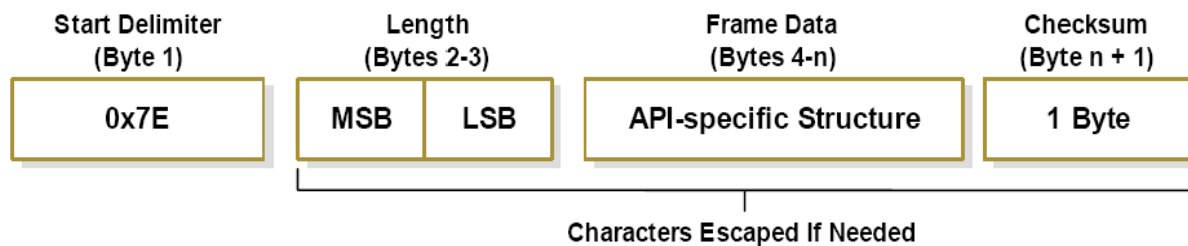


Figura 29. Trama de datos enviada y recibida en modo API.[Digi]

El objetivo del primer ensayo de esta segunda fase de desarrollo ha sido la realización de una comunicación entre tres módulos emisores (*end device*) con dos entradas analógicas activas cada uno de ellos (a cada entrada analógica le corresponde un sensor) y un coordinador. Cuando se produce una variación en la medida de alguno de los sensores de cualquiera de los equipos, se envía la nueva medida al coordinador que se encarga de enviarle al PC los datos correspondientes a la variación para que sean representados por pantalla.

Esta comunicación será unidireccional, es decir, sólo existe envío de datos de cada emisor al coordinador.

El funcionamiento de la red es exactamente igual que en el de la fase uno de desarrollo. Cada *end device* envía al coordinador la lectura de los diferentes sensores que a él van unidos (dos por equipo) cuando se produce una variación en la medida de datos éstos son representados en la interfaz gráfica.

Cada *end device* manda una trama de datos al coordinador como la que aparece en la Figura 30. Donde como datos le envía: el canal de lectura, es decir, si los datos proceden del sensor uno o del sensor dos, si se está por encima del valor máximo de tensión, 5 Voltios, y el valor de la medida realizada por el sensor.

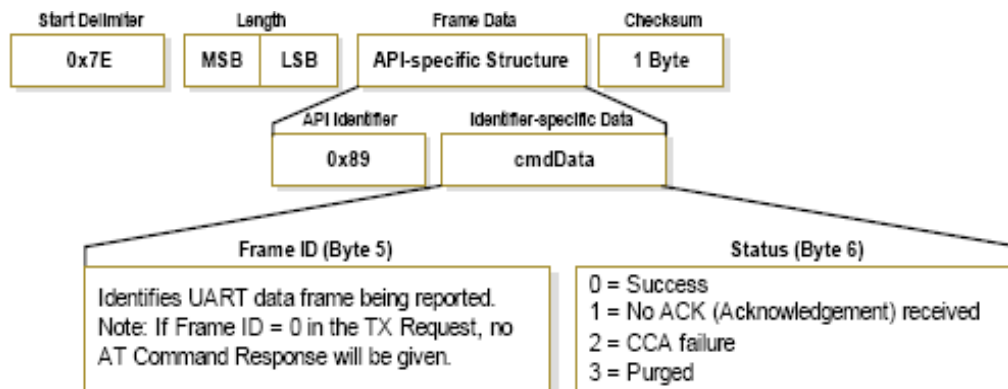


Figura 30. Trama de datos enviada por cada emisor.[Digi]

El coordinador, a su vez, envía la trama recibida al PC, que será como la que aparece en la Figura 31, para que la interfaz gráfica muestre por pantalla el valor de cada sensor.

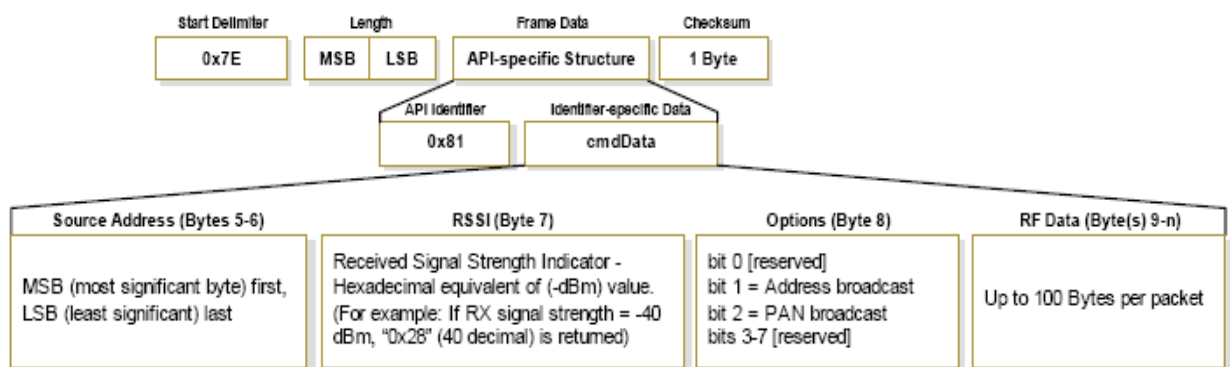


Figura 31. Trama de datos recibida por el coordinador.[Digi]

Al ser una comunicación unidireccional, el coordinador no puede enviar datos a ninguno de los *end device*. Esto es debido a que tanto en la fase uno de desarrollo como en la fase dos, se utiliza el microprocesador Atmega de la placa Arduino para procesar los datos. Para ello, los *jumpers* de modo

de comunicación en la placa Arduino-coordinador están dispuestos en modo XBee, esto hace que no sea posible la recepción de datos desde el PC.

Para poder realizar una comunicación bidireccional será necesario eliminar el microprocesador de la parte Coordinadora, haciendo que sea el programa diseñado para la interfaz gráfica quien tramite los datos recibidos de los distintos *end device* y a su vez sea el encargado de generar la trama de datos necesarios cuando se quiera enviar algún mensaje desde el coordinador a algún *end device*. También se necesita cambiar la disposición de los *jumpers* de comunicación y colocarlos en modo USB para que todo lo que se reciba por el XBee del coordinador pase al PC.

Como se ha explicado anteriormente, el sistema desarrollado en esta fase, tiene una comunicación unidireccional, Emisores → Coordinador → PC. Con ella, se ha conseguido satisfactoriamente la comunicación buscada y por tanto, poner en funcionamiento una red punto a multipunto con coordinador.

Sobre la aplicación existente en la fase uno de desarrollo se introducen algunas mejoras como las que se pueden observar en la Figura 32.

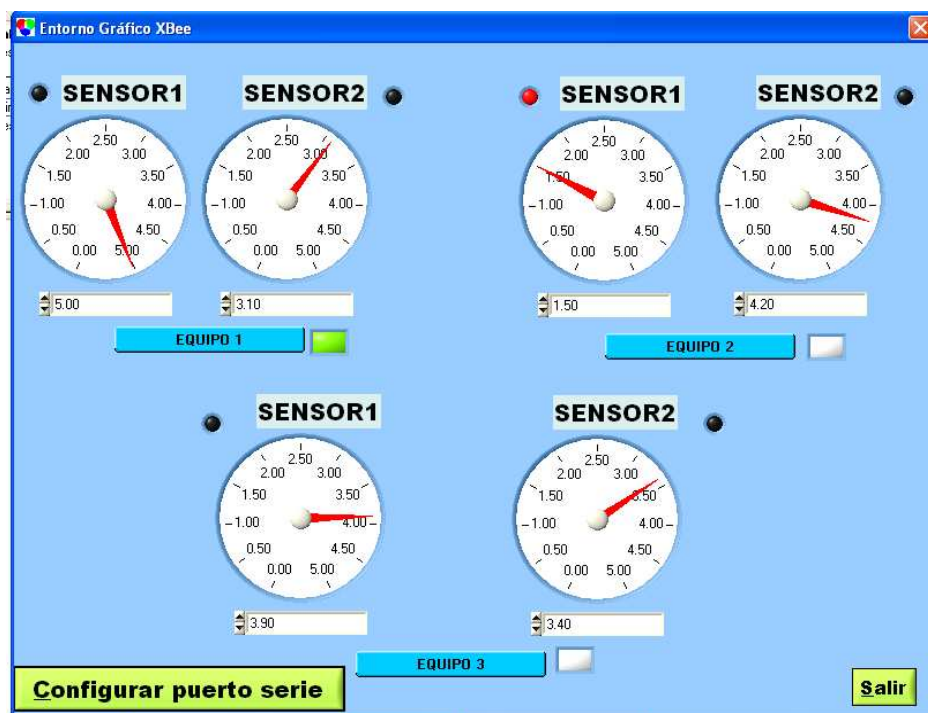


Figura 32. Aplicación realizada al finalizar la segunda fase de desarrollo.



El diseño final de la pantalla donde aparecen representados las diferentes medidas de los seis sensores que forman la red sensorial cuenta con:

- Seis relojes de medida, dos por equipo emisor, y cada equipo representa cada uno de los sensores del sistema.
- Seis LEDs de color rojo sobre cada reloj que se ilumina cuando se produce una variación en el valor de medida de cada sensor.
- Tres LEDs verdes, uno por equipo, que permanece encendido cuando alguno de los sensores sobrepasa los 5 Voltios establecidos como límite máximo. Se puede decir que hace de alarma.
- Los botones de configuración del puerto serie de comunicación y el botón de salir de la aplicación.

Con estas mejoras introducidas en la interfaz gráfica se ha buscado obtener una mejor visualización de las variaciones producidas en las medidas y una rápida apreciación de que se ha producido una sobretensión en alguno de los equipos.

La Tabla 6 recoge las principales características de la red punto a multipunto llevada a cabo en esta segunda Fase de trabajo.

CARACTERÍSTICAS PRINCIPALES DE LA FASE II	
	FASE II
Tipo de red	Punto a Multipunto
Tipo de comunicación	Unidireccional
Nº kits Arduino en la red	4
Nº <i>end devices</i>	3
Nº <i>end devices</i> máx. que soporta la red	65.635
Nº sensores por <i>end device</i>	2
Control de envío y recepción de datos	SÍ
Modo programación módulos	API
Microprocesador coordinador	SÍ
Disposición JUMPERS de comunicación en coordinador	XBee

Tabla 6. Características de la red punto a multipunto desarrollada en la Fase II.

6.3. Tercera fase de desarrollo del proyecto.

Al comienzo de la tercera fase se cuenta con una red punto a multipunto con coordinador y comunicación unidireccional como se ha explicado anteriormente. En esta parte del proyecto se desea implantar una comunicación bidireccional dentro de la red existente para así poder contar con un sistema más robusto y a prueba de fallos como el que se muestra en la Figura 33.

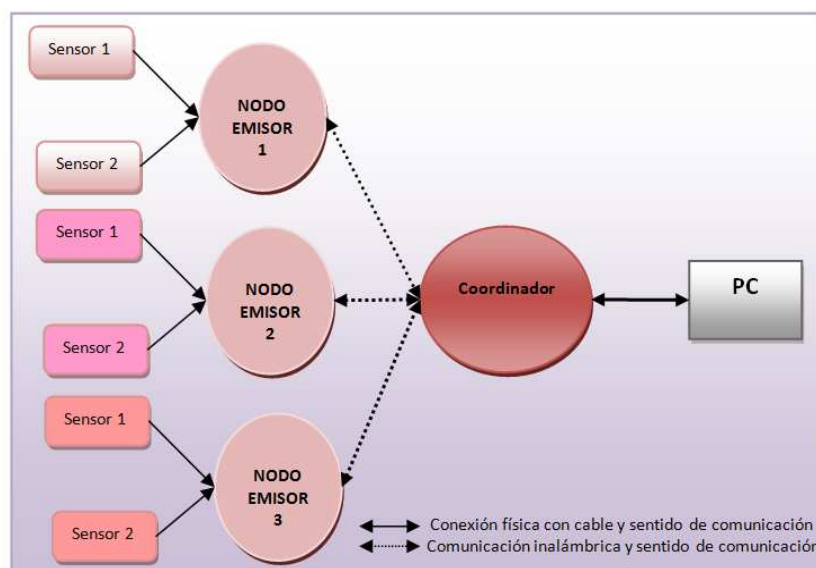


Figura 33. Sistema diseñado en la Fase III.

La comunicación bidireccional, *End device* \leftrightarrow Coordinador \leftrightarrow PC, brinda al usuario la posibilidad de realizar un testeo de los sensores siempre que lo desee para así verificar que el sistema está funcionando correctamente y las medidas representadas por pantalla son las medidas actuales de cada sensor.

El equipo Arduino-coordinador sufre algunas modificaciones en esta parte, el microprocesador de la parte Coordinadora es eliminado, haciendo que sea el programa diseñado para la interfaz gráfica quien tramite los datos recibidos de los distintos *end device* y a su vez sea el encargado de generar la trama de datos necesarios cuando se quiera enviar algún mensaje desde el coordinador a algún *end device*. También se necesita cambiar la disposición de los *jumpers* de comunicación y colocarlos en modo USB para que todo lo que se reciba por el XBee del coordinador pase al PC.

La forma de analizar los datos recibidos por la aplicación de la interfaz gráfica varía, ahora se debe analizar la trama de recepción que el XBee del coordinador manda al PC para comprobar que la recepción ha sido correcta.

La pantalla de representación de datos que muestra la Figura 34 es la misma que la utilizada durante el último ensayo de la fase dos de desarrollo a excepción de que ahora se incorpora un nuevo botón en dicha pantalla. El nuevo botón, TESTEO, es el que da la posibilidad al usuario de realizar una comprobación de que los datos mostrados antes de pulsarlo son los mismos que después de que ha sido presionado.

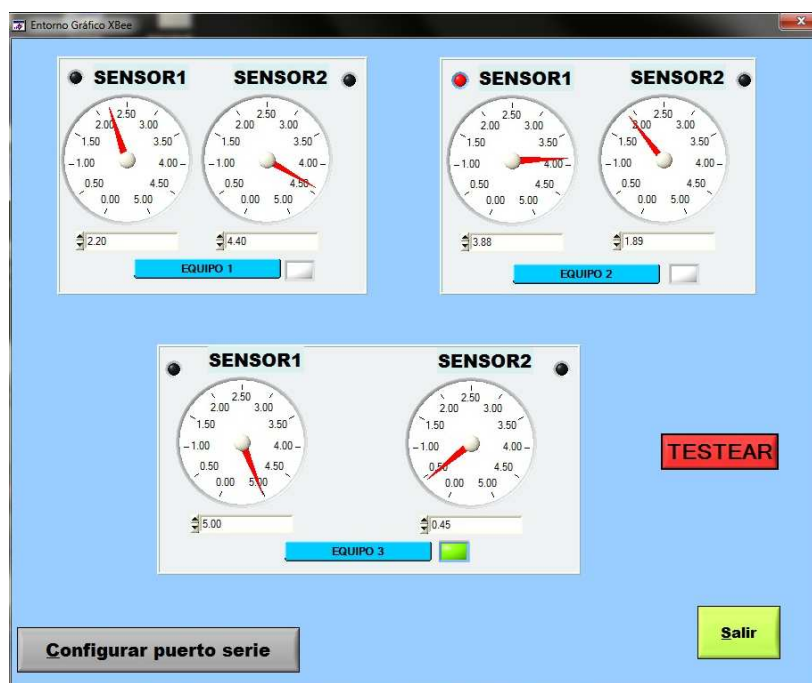


Figura 34. Pantalla de la aplicación final diseñada.

La función de testeo se encarga de enviar la trama de datos correspondiente al envío de datos a cada dispositivo *end device* para que cada uno de ellos haga una lectura de cada uno de los dos sensores que tienen unidos y le devuelva al coordinador el valor de la nueva medida.

Otra de las mejoras introducidas en la aplicación gráfica ha sido la creación de un fichero .txt donde se recogen todos los datos de medida de nuestro sistema. Dicho fichero se guarda en la carpeta

donde se encuentra el ejecutable de la interfaz gráfica. En la Figura 35 aparece el fichero creado, cuyo nombre es data.txt. El archivo cuenta con cinco columnas:

- Tiempo. Recoge la hora en la que se produce una nueva medida.
- Equipo. Indica el nodo emisor que envía los datos.
- Sensor. Salida analógica a la está unido el sensor que ha variado.
- Lectura. Datos de medida.
- Alarma. Se pone a '1' cuando el valor monitorizado está por encima de 5V.



TIEMPO	EQUIPO	SENSOR	LECTURA	ALARMA
11:04:51	2	1	3,94	0
11:04:53	3	1	5,00	1
11:04:54	2	1	3,93	0
11:04:55	3	1	0,81	0
11:04:57	3	0	0,33	0
11:03:23	3	0	0,22	0
11:03:24	3	1	0,78	0
11:03:28	3	0	1,73	0
11:03:30	3	0	0,10	0
11:03:32	3	1	1,62	0
11:03:33	3	0	0,09	0
11:03:34	3	1	5,00	1
11:03:39	3	1	5,00	1
11:06:27	3	1	2,55	0
11:06:29	3	1	3,14	0
11:06:31	3	0	2,59	0
11:06:33	3	0	2,57	0
11:06:37	3	1	3,13	0

Figura 35. Fichero data.txt que recoge los datos monitorizados.

Con estas últimas mejoras introducidas en la red se ha conseguido obtener:

- Una mejora en la visualización de las variaciones producidas en las medidas y una rápida apreciación de que se ha producido una sobretensión en alguno de los equipos.

- Fiabilidad de las medidas representadas por pantalla.
- Un Sistema más robusto y con posibilidades de incorporación de mejoras y de diferentes aplicaciones.

Tras obtener la red sensorial buscada en esta Fase III, se procede a hacer unas pruebas para verificar el alcance de la red desplazando el nodo tres utilizado para medir hasta donde llega la señal. Para ello, se desplaza dicho nodo emisor por diferentes puntos comprobando su alcance y donde la red deja de recibir correctamente la información. En la Figura 36 aparece el plano de la planta donde se encuentra el nodo coordinador y los diferentes puntos por donde se ha trasladado el nodo emisor.

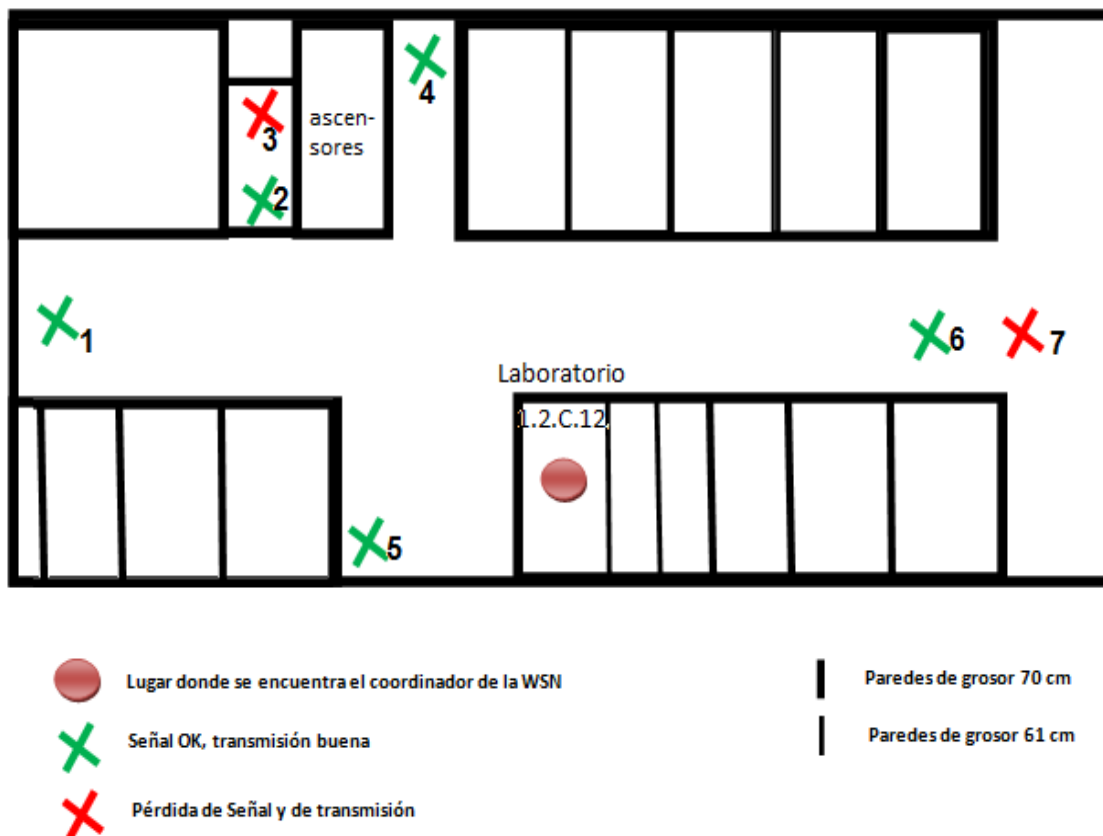


Figura 36. Plano de los puntos donde se ha comprobado el alcance.

En la Tabla 7 se recoge la distancia de cada uno de los puntos donde se ha comprobado el alcance de la red ante la existencia de paredes de diferentes dimensiones, comprobando que al meter el dispositivo en el punto tres, se pierde la transmisión de datos a una distancia de 17 metros, debido a la existencia de paredes muy gruesas, con una puerta de por medio y los ascensores. Sin embargo en el punto 7 también existen muchas paredes de por medio pero la transmisión continúa hasta una distancia de 28 metros. Según las especificaciones del fabricante, el alcance de estos módulos ante la existencia de paredes es de unos 30 metros, por lo que nuestros módulos se aproximan a estos valores.

DISTANCIAS DEL NODO EMISOR AL NODO COORDINADOR		
PUNTO EN EL PLANO	DISTANCIA AL NODO COORDINADOR	FALLO EN LA TRANSMISIÓN
1	26 m	NO
2	14 m	NO
3	17 m	SÍ
4	19 m	NO
5	10 m	NO
6	28 m	NO
7	29 m	SÍ

Tabla 7. Alcance de la red.

En el anexo 6 se encuentra todo el código fuente utilizado en cada equipo *end device* y en la aplicación gráfica desarrollada en esta fase final.

La Tabla 8 recoge las principales características de la red punto a multipunto llevada a cabo en esta tercera Fase de trabajo.

CARACTERÍSTICAS PRINCIPALES DE LA FASE III	
	FASE III
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Nº kits Arduino en la red	4
Nº end devices	3
Nº end devices máx. que soporta la red	65.635
Nº sensores por end device	2
Control de envío y recepción de datos	SÍ
Modo programación módulos	API
Microprocesador coordinador	NO
Disposición JUMPERS de comunicación en coordinador	USB
Alcance de la red (con paredes según nuestras medidas)	17 - 28 m
Alcance de la red (con paredes según fabricante)	30 m
Alcance de la red (sin paredes según fabricante)	100 m
Potencia consumida módulo XBee RF (según especific.)	≤ 155 mW
Potencia consumida nodo coordinador	350 mW
Potencia consumida nodo emisor 1	965,43 mW
Potencia consumida nodo emisor 2	1227,7 mW
Potencia consumida nodo emisor 3	627 mW
Potencia consumida dispositivo emisor 1	410 mW
Potencia consumida dispositivo emisor 2	550 mW
Potencia consumida dispositivo emisor 3	627 mW
Potencia consumida emisor HFBR – 1528 sensor pulido Nivel	271,65 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir Nivel	283,3 mW
Potencia consumida fotodetector IF-D91 sensor pulido Nivel	0,165 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir Nivel	0,315 mW
Potencia consumida emisor HFBR – 1528 sensor pulido Tª	328,4 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir Tª	348,8 mW
Potencia consumida fotodetector IF-D91 sensor pulido Tª	0,016 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir Tª	0,555 mW

Tabla 8. Características de la red punto a multipunto desarrollada en la Fase III.

Las potencias consumidas en los dispositivos Arduinos que recoge la Tabla 8, correspondientes con el nodo coordinador y los nodos emisores, se han obtenido a partir de la tensión de alimentación (5 V) y de la intensidad que circula a la entrada de cada módulo.

Con la ayuda de un amperímetro y la fabricación de un cable USB como el que aparece en la Figura 37 se ha obtenido la medida de intensidad de los Arduinos. Este cable va unido mediante su extremo USB B a la placa Arduino, y mediante su extremo conector de USB B al cable de alimentación USB A-B (que está enchufado al PC). Entre los cables de alimentación del cable fabricado (de color rojo) se coloca el amperímetro encargado de monitorizar la medida de intensidad.

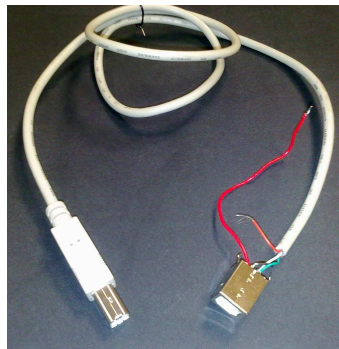


Figura 37. Cable USB B-B fabricado para medir la intensidad a la entrada de los Arduinos.

Las potencias consumidas por los emisores ópticos se han obtenido también a partir de la tensión de alimentación (5 V) y de la intensidad que circula por cada elemento. Esta intensidad ha sido medida mediante la colocación de un amperímetro en serie en cada uno de los elementos.

En el caso de la potencia consumida por el módulo de RF XBee en la comunicación ZigBee, según el fabricante de la placa Arduino, no puede ser mayor a 132 mW, ya que la alimentación del dispositivo XBee de 3,3 V es generada por el chip FTDI integrado en la placa y la máxima corriente que soporta dicho chip es de 50mA.

Según las especificaciones, recogidas en capítulos anteriores, de las distintas comunicaciones inalámbricas, la comunicación ZigBee tiene un consumo de unos 35 mA en la transmisión de datos, a diferencia de los 400 mA que consume WiFi.



Para conocer la potencia consumida total por cada nodo emisor, es decir, la formada por la suma de la potencia consumida por su dispositivo Arduino, las potencias consumidas por sus emisores ópticos y las potencias consumidas por su fotodetectores se debe utilizar la fórmula (2):

$$P_{\text{nodo emisor}}(i) = P_{\text{dispositivo emisor}}(i) + P_{\text{emisores ópticos}}(i) + P_{\text{fotodetectores}}(i) \quad (2)$$



Capítulo 7. *Software.*

El Entorno de Desarrollo de Arduino (del inglés IDE - *Integrated Development Environment*) es una herramienta que funciona en diferentes plataformas (Windows, Macintosh, Linux) dedicada a la edición y descarga de programas en la placa Arduino. El código que se ejecuta en las placas está basado en las librerías de aplicación Wiring API y en las de desarrollo de los integrados AVR. Utilizando un *software* gratuito que se puede adquirir en la página web de Arduino. **[Arduino]**

Para configurar los módulos XBee el *software* utilizado se llama XCTU que como ocurre con el *software* de Arduino también es gratuito, y con él se configurarán los módulos XBee, su velocidad de transmisión, la dirección propia de cada módulo y la del módulo con el que se comunican, etc.

El entorno gráfico ha sido diseñado con LabWindows/CVI de Nacional Instruments que es un ambiente de desarrollo completamente integrado, especialmente diseñado para la creación de sistemas de instrumentación basados en GPIB, PXI, VXI, tarjetas de adquisición de datos insertables y

otros. Se combina un interactivo de desarrollo de fácil uso con el poder y flexibilidad de programación de código compilado en ANSI C. LabWindows/CVI proporciona muchas características útiles que incrementan su productividad sin sacrificar la velocidad o fácil manipulación del código fuente compilado en C.

7.1. Flujogramas del código fuente.

En los siguientes flujogramas, se puede observar el *software* diseñado en las distintas pruebas realizadas, que se han explicado anteriormente:

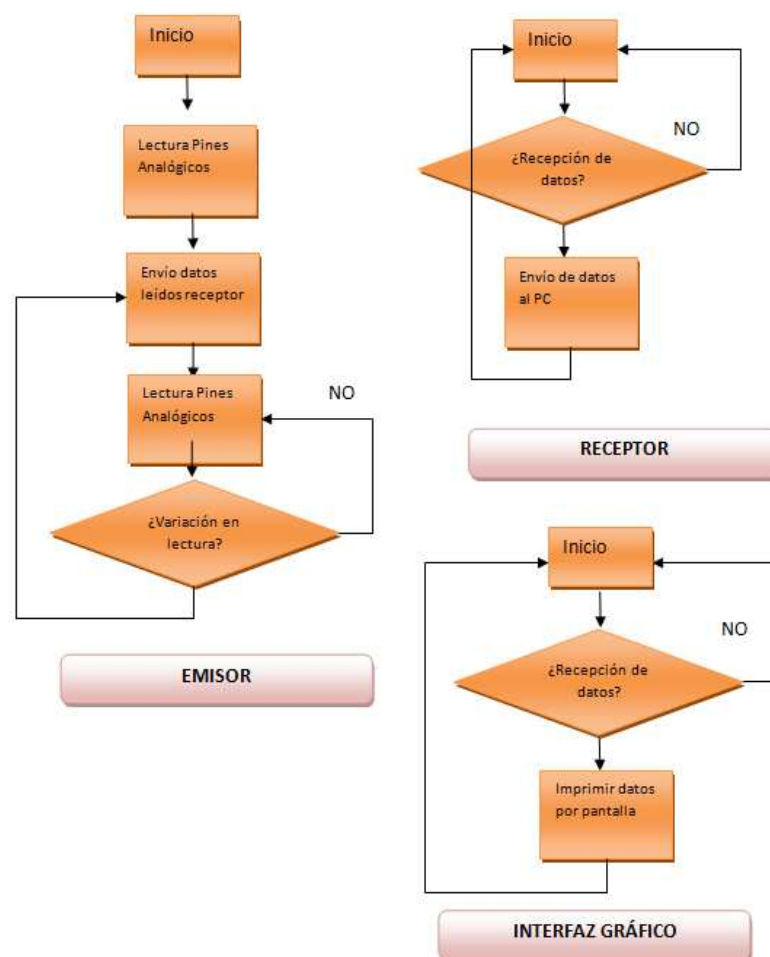


Figura 38. Flujograma de la red punto a punto de la Fase I.

En la Figura 38 aparece el diagrama de flujo del primer ensayo realizado, es decir, de la red punto a punto. Se tienen tres flujogramas, uno correspondiente al equipo emisor que se encuentra a la espera de que se produzca una variación en una entrada analógica para enviar los datos al receptor.

El dispositivo receptor está a la espera de recibir esos datos para enviarlos al ordenador y que la interfaz gráfica los saque por pantalla.

En la Figura 39 se hace referencia al diagrama de flujo del segundo ensayo realizado, el diseño de una red con coordinador, donde la diferencia con lo explicado anteriormente es que tanto el envío como la recepción de datos debe realizarse siguiendo una trama de envío y de recepción. Si la trama no es correcta, no se produce la recepción en el equipo coordinador.

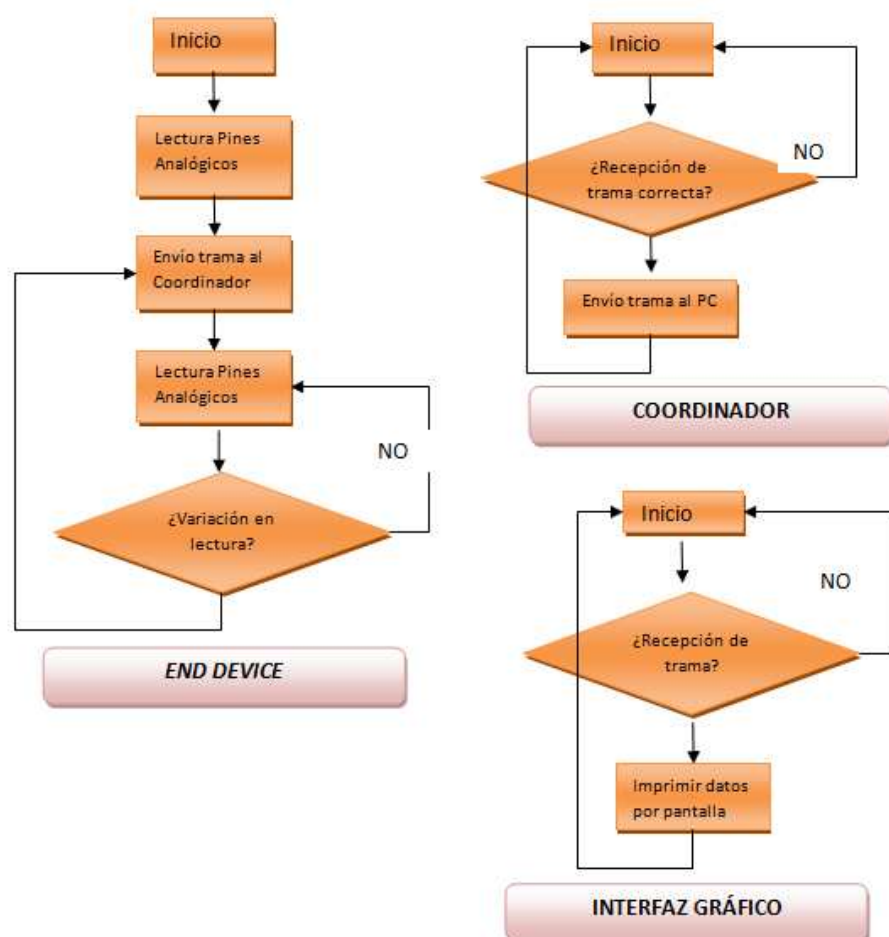


Figura 39. Flujograma de la red punto a multipunto de la Fase II.

En la Figura 40 se hace referencia al diagrama de flujo del tercer ensayo realizado, el diseño de una red con coordinador con comunicación bidireccional, donde cada *end device* realiza la lectura de los sensores que a él van unidos, cuando se produce una variación envía la trama de datos al coordinador, y si el *end device* recibe una trama de datos del coordinador, realiza una nueva lectura y envío de datos de los dos sensores con los que cuenta. Estos datos son recogidos en un fichero de texto llamado data.txt cada vez que se produce una variación de medida.

La aplicación gráfica ahora cuenta con el envío de una trama de datos por el puerto serie al coordinador cada vez que el usuario presione el botón de testeo de la pantalla principal.

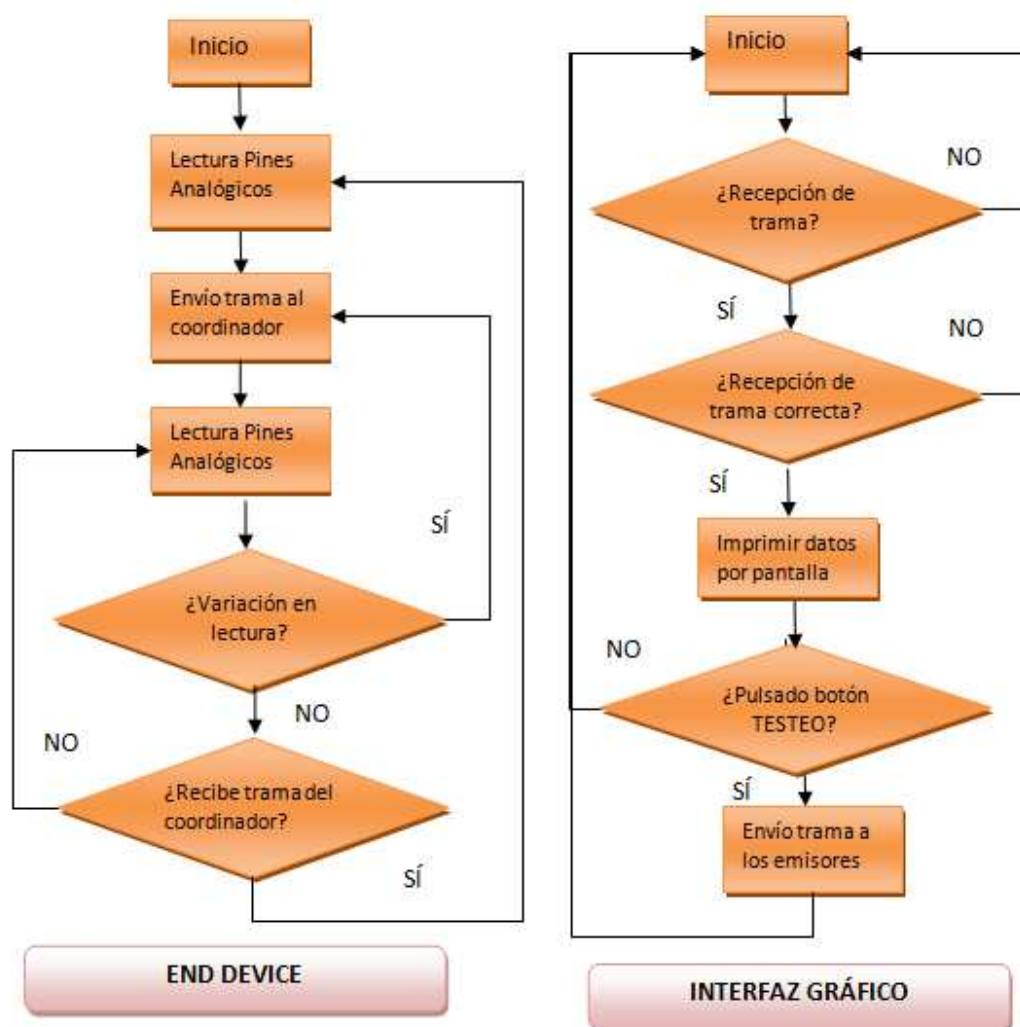


Figura 40. Flujograma de la red punto a multipunto de la Fase III.

7.2. Calibración de los módulos Arduino emisores.

Debido a las dos posibilidades de alimentación de los módulos Arduino, mediante puerto USB o con un cargador de red, y a las pequeñas variaciones en la tensión suministrada entre diferentes USB, es necesario realizar una calibración de los dispositivos Arduinos en función de dónde van a ir conectados.

Esta calibración es necesaria por la conversión analógica-digital de 10 bits que se produce al realizar una lectura de los datos de entrada de cada sensor.

La no linealidad del conversor, y su resolución de 1LSB = 4 mV junto con las variaciones que se producen en la alimentación del Arduino dependiendo del lugar donde se conecte requiere que al valor obtenido de la conversión mediante la fórmula (3) que nos da su fabricante, se le sume un factor para hacer coincidir el valor de entrada al conversor con el valor de salida y así corregir la desviación que se produce.

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}} \quad (3)$$

El factor de corrección, puede variar dependiendo de cómo se alimente al módulo Arduino o del puerto USB donde se conecte la alimentación. Para saber su valor exacto es necesario recoger varias medidas entre al valor de conversión analógica-digital que se obtiene y la medida real. Con ello se podrá estimar el valor de dicho factor, y comprobar que para un mismo valor de entrada, el valor de la conversión es diferente.

El código fuente programado para hacer esta comprobación es el siguiente:

```
#include <avr/io.h>
int analogPin0 = 0;
int analogPin1 = 1;
int analogPin2 = 2;
int analogPin3 = 3;
int analogPin4 = 4;
int analogPin5 = 5;
```



```
int b2, b3;int array[10];

void setup(){
  Serial.begin(9600);
}

void loop()
{
  leerPIN0();
}

void leerPIN0(void)
{
  int i;
  b2=0;
  delay(3000); // toma de medidas cada tres segundos para poder anotar los datos

  for(i=0;i<10;i++){
    array[i]=analogRead(analogPin0);
    b2=b2+array[i];
    delay(25);
  }
  b2=b2/10; // Resultado de la conversión

  b3=(b2*0.4); // Resultado de la fórmula (3), donde posteriormente se le sumará el factor de
  corrección.

  Serial.print("\n Valor conversión ");

  Serial.print(b2,DEC);

  Serial.print("\n Valor de la entrada analógica tras la conversión= ");

  Serial.print(b3,DEC);Serial.print("\n\n\n ");

}
```

Con este código fuente programado en cada uno de los módulos, que se encontrarán alimentados y conectados en el lugar final donde irán colocados se recogen las medidas del valor de la conversión y el valor de la entrada analógica después de la conversión.

A continuación se muestran las tablas recogidas con la calibración de los módulos Arduinos utilizados en este proyecto. Se han calibrado conectándolos a la red con un cargador, y a diferentes puertos USB de dos ordenadores.

La columna resultado ADC es el resultado decimal de la conversión analógica digital. La V_{in} conversor se corresponde con el valor de la tensión de entrada al conversor medida con un polímetro, siendo V_{out} conversor el valor obtenido tras realizar la conversión utilizando la fórmula (3). El factor de corrección se obtiene de la relación que se produce entre el resultado del ADC y la variación entre V_{in} y V_{out} .

La Tabla 9 se corresponde con la calibración de un nodo emisor alimentado mediante el USB delantero del PC de sobremesa utilizado, su factor de corrección sería 19.

Resultado ADC	V_{in} conversor[V]	V_{out} conversor[V]	$(V_{in} - V_{out})[V]$	Factor
612	2,76	2,45	0,31	19
669	3,01	2,67	0,34	19
702	3,16	2,80	0,36	19

Tabla 9. Calibración Arduino conectado al USB delantero de un PC de sobremesa.

La Tabla 10 pertenece a un equipo emisor alimentado al USB trasero del mismo PC, con un factor de corrección de 12.

Resultado ADC	V_{in} conversor[V]	V_{out} conversor[V]	$(V_{in} - V_{out})[V]$	Factor
448	2,17	1,79	0,38	12
648	3,13	2,59	0,54	12
861	4,16	3,44	0,72	12

Tabla 10. Calibración Arduino conectado al USB trasero de un PC de sobremesa.

La Tabla 11 recoge los datos de un emisor alimentado mediante el USB de un ordenador portátil con un factor de corrección de 36.

Resultado ADC	Vin conversor[V]	Vout conversor[V]	(Vin - Vout)[V]	Factor
655	2,80	2,62	0,18	36
704	3,01	2,81	0,20	36
955	4,08	3,82	0,26	36

Tabla 11. Calibración Arduino conectado al USB de un PC portátil.

En la Tabla 12 se tienen los datos de un nodo emisor enchufado a la red mediante un cargador, cuyo factor de corrección es 11.

Resultado ADC	Vin conversor[V]	Vout conversor[V]	(Vin - Vout)[V]	Factor
615	3,01	2,46	0,55	11
853	4,18	3,41	0,73	11
917	4,49	3,67	0,82	11

Tabla 12. Calibración Arduino conectado a la red mediante un cargador.

Se puede comprobar que en tres de los casos anteriores, para una misma medida de 3,01 voltios, el valor de la conversión analógica digital es diferente, por ello la necesidad de este factor de corrección.

Una vez conocido el factor de corrección, se puede decir que los módulos Arduinos están calibrados con relación al lugar donde van a ser alimentados. En el código fuente de cada módulo emisor se debe modificar la línea de código que hace referencia al resultado de la conversión incluyendo el valor del factor de cada caso. Esta línea de código en la función “leerPIN” es la siguiente:

$$b3=(b2*0.4)+(b2/FACTOR\ DE\ CORRECCIÓN);$$



7.3. Ampliación de la Red Sensorial.

La aplicación gráfica diseñada está preparada para recibir y enviar datos a tres nodos emisores con dos sensores por nodo, que es lo que se representa en la pantalla de monitorización de datos.

Si en un futuro se quisiera ampliar la red sensorial, la introducción de nuevos nodos emisores no influye a los dispositivos existentes en la red, es decir, cada módulo es independiente de los otros.

Cada nuevo módulo XBee (perteneciente a cada nuevo nodo emisor) debe ser programado mediante el software XCTU habilitándose como dispositivo final, con su dirección, la dirección del coordinador y el modo API. Como en la trama de datos enviada por el emisor al coordinador incluye el identificador del equipo que le envía los datos, el módulo coordinador no necesita ningún cambio de programación.

Si se quiere ampliar el número de sensores por nodo, el código fuente programado mediante el software de Arduino para cada dispositivo final para leer la información de los dos sensores de cada nodo debe variar, incluyéndose en él la lectura y representación de las diferentes entradas analógicas a las que se les una un nuevo sensor. Habría que introducir una nueva función como correspondiente a la entrada analógica cero “void leerPIN0(void)”, cambiando los subíndices de valor cero por el subíndice de la nueva entrada. El código fuente se recoge en el Anexo 6, y ahí puede verse la función leerPIN0 anteriormente mencionada.

Sin embargo, la interfaz gráfica se vería más afectada ante la introducción de nuevos dispositivos finales o ante la introducción de más sensores por nodo. Sería necesario cambiar la apariencia de la interfaz, introduciendo más relojes de representación de medida con su correspondiente código de programación. Si el número de nuevos nodos o nuevos sensores es muy elevado, habría que rediseñar la apariencia de la monitorización de datos por pantalla, de forma que se pueda abarcar la representación de todos los datos. Se podría diseñar una pantalla principal donde sólo aparezcan botones de apertura de ventanas y LEDs de alarma para avisar de cambios en los nodos. Cada botón iría asociado a una pantalla donde se encuentren representados los relojes que monitorizan la información de cada sensor, y así el usuario pueda abrir el nodo que en cada momento desee visualizar. Dentro de estas modificaciones existen multitud de posibilidades de rediseño.



En los Anexos de este proyecto se recoge el código fuente correspondiente a cada ensayo y los diferentes programas utilizados durante este proyecto, X-CTU, Arduino y LabWindows. En dichos anexos se contempla de una forma más detallada como funciona cada software, para qué se ha utilizado y la forma de utilizarlo.

7.4. Requisitos del *software* utilizado.

El *software* utilizado para el desarrollo de este proyecto se encuentra instalado en el PC del laboratorio 1.2.C.12 cuyo número de inventario se corresponde con 197356.

El XCTU se descarga de forma gratuita de la página web de DIGI, y se instala normalmente en el ordenador.

El *software* de Arduino se descarga también gratuitamente de su misma página y dentro de la carpeta se encuentra en modo portable, por lo que no es necesaria su instalación en el PC.

EL programa utilizado para la interfaz gráfica, el LabWindows necesita una instalación previa y la adquisición de su licencia. Es recomendable que los proyectos realizados en esta plataforma incluyan todas las librerías necesarias dentro de la carpeta donde se encuentra el proyecto, pudiendo realizar posteriormente un ejecutable de dicho proyecto. Para que ese ejecutable funcione en equipos donde el programa no está instalado se requiere de un *runtime* de LabWindows.

El fichero de texto que recoge las medidas representadas en la aplicación gráfica se guarda en la carpeta donde se encuentra el ejecutable del programa.



Capítulo 8. Diseño final.

Tras la ejecución de este proyecto se cuenta con una red inalámbrica de sensores de fibra óptica de plástico con tecnología ZigBee, donde se han empleado:

- Cuatro Kits Arduino- Diecimila. Tres utilizados como nodos emisores y uno como coordinador.
- Tres microprocesadores Atmega168 colocados en cada dispositivo emisor.
- Tres cables USB A-B y un cargador de red.
- Tres fuentes de alimentación.
- Cuatro sensores de fibra óptica de plástico con sus respectivos circuitos de acondicionamiento.

En este capítulo se van a mostrar las diferentes partes que forman la WSN con sus respectivos componentes, los resultados obtenidos y el montaje final realizado.

Según el diagrama de bloques de la Figura 41 la red de sensores inalámbrica desarrollada en este proyecto puede dividirse en los siguientes bloques:

- Nodo receptor.
- Nodos emisores.
- Sensores de Temperatura.
- Sensores de nivel.

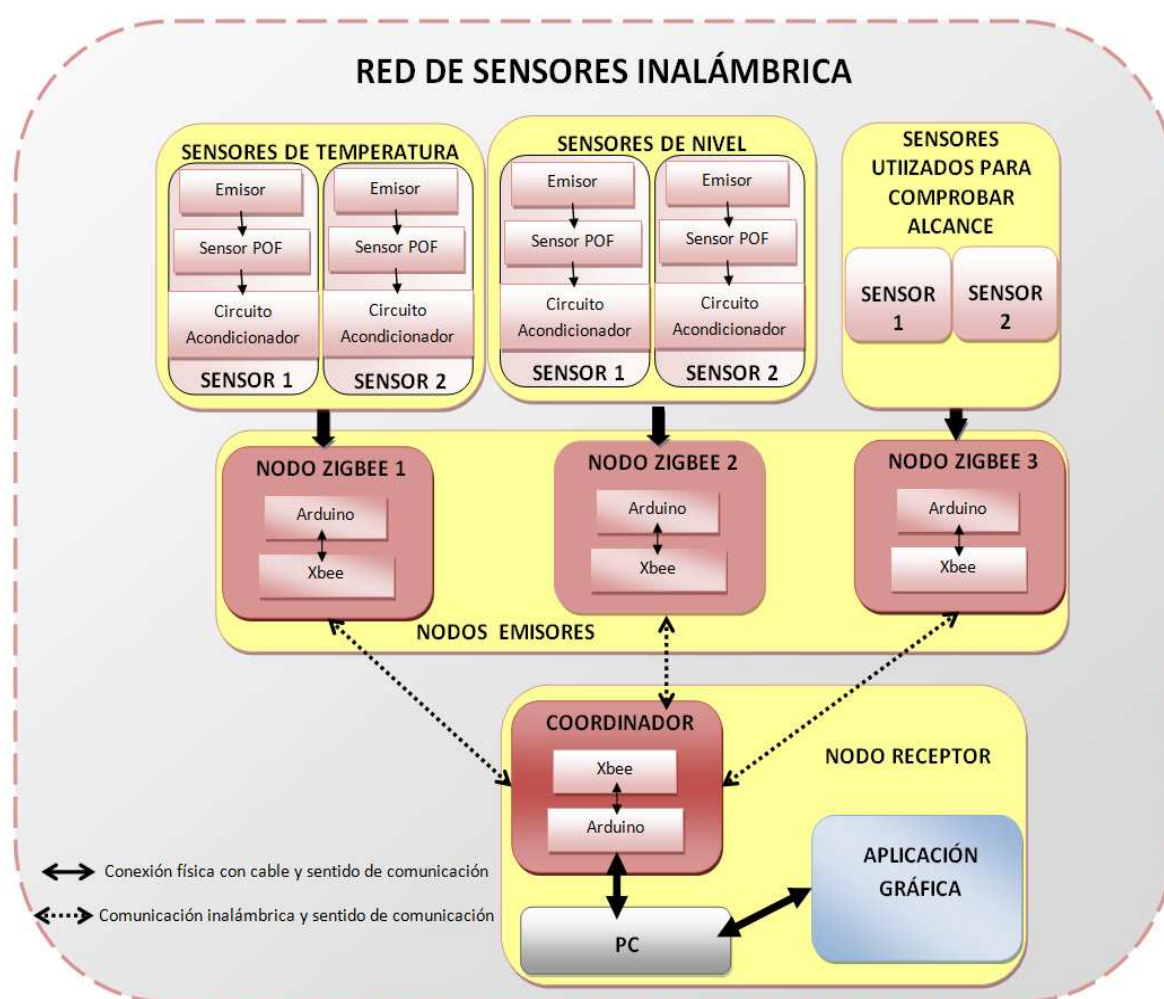


Figura 41. Diagrama de bloques de los componentes de la WSN final.

Se puede decir que el conjunto de todos los bloques mencionados anteriormente dan lugar a una WSN con tecnología ZigBee y sensores de fibra óptica de plástico.

A continuación, se puede ver en la Figura 42 el montaje de la red sensorial realizado en el laboratorio de pruebas. Aquí aparecen los dos nodos sensores de Arduino, los sensores de POF y sus circuitos de acondicionamiento.

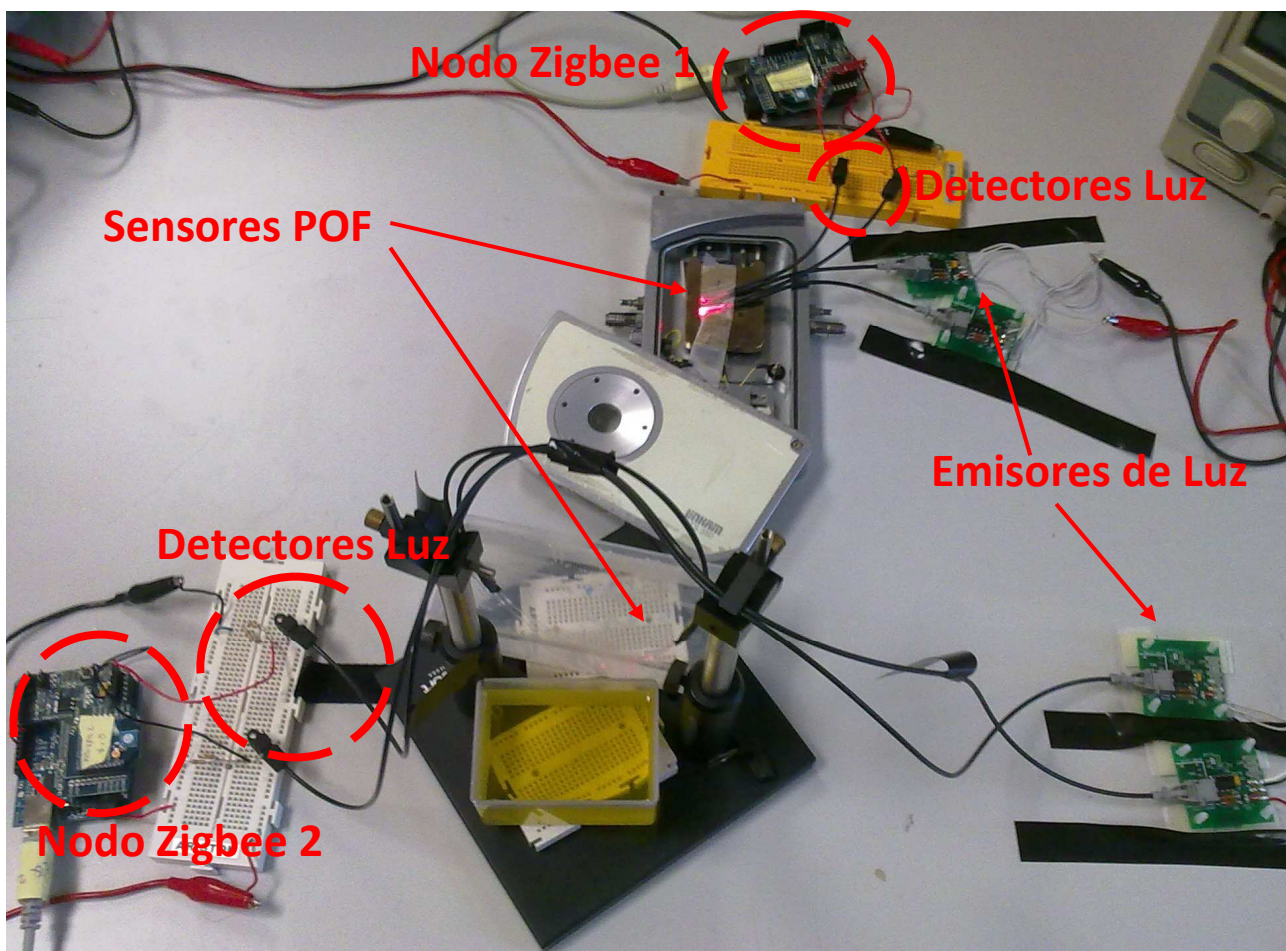


Figura 42. Montaje final realizado.

En la Figura 43 , aparece el nodo sensor utilizado en esta red para comprobar la distancia a la que se puede colocar dicho nodo ante la existencia de obstáculos en la transmisión y que la comunicación entre equipos siga existiendo. Ha sido colocado en otra sala del laboratorio y en diferentes puntos

del pasillo, simulando las entradas sensoras con una fuente de alimentación, para facilitarnos la movilidad del sistema.

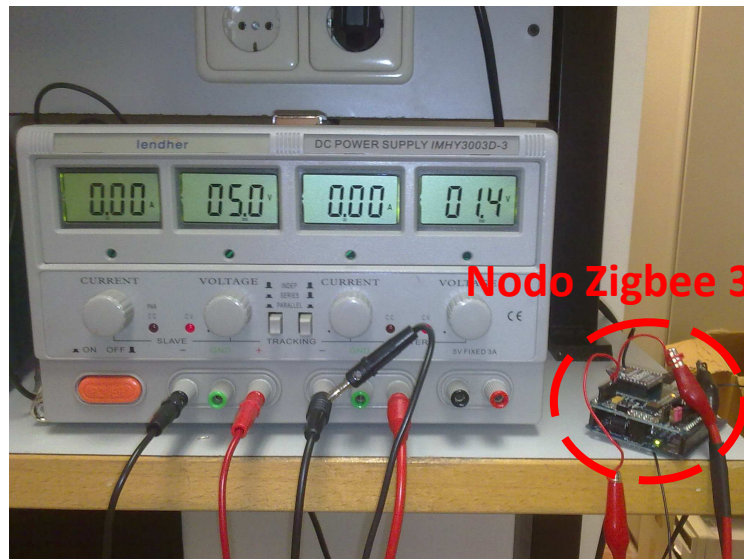


Figura 43. Nodo sensor utilizado para verificar la transmisión de datos ante la existencia de paredes.

El coordinador, es un Arduino unido al PC mediante un cable USB como se puede ver en la Figura 44.



Figura 44. Coordinador unido al PC.

8.1. Red de sensores inalámbrica.

Como se explica en el primer capítulo de este documento, una WSN está formada por numerosos dispositivos (llamados nodos) que utilizan sensores distribuidos espacialmente en diferentes puntos y comunicándose entre sí de manera inalámbrica, se encargan de analizar información como temperatura, nivel, presión, etc. en zonas específicas.

La red sensorial diseñada está formada por tres nodos sensores y un nodo que hace de estación central encargado de monitorear datos sobre temperatura y nivel y se comunican entre sí mediante el protocolo de comunicación 802.15.4., en la Tabla 13 se recogen las especificaciones de la WSN.

ESPECIFICACIONES DE LA RED SE SENSORES INALÁMBRICA	
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Estándar de comunicación	802.15.4.
Nº kits Arduino en la red	4
Nº <i>end devices</i>	3
Nº <i>end devices</i> máx. que soporta la red	65.535
Nº sensores por <i>end device</i> utilizados	2
Nº sensores por <i>end device</i> posibles	6
Control de envío y recepción de datos	SÍ
Modo programación módulos	API
Microprocesador Coordinador	NO
Disposición JUMPERS de comunicación en Coordinador	USB
Ancho de Banda	250 kbps
Alimentación módulos Arduino	5V
Alimentación emisores HFBR – 1528	5 V
Alimentación fotodetector IF-D91	5 V
Potencia consumida nodo receptor	350 mW
Potencia consumida nodo emisor 1	965,43 mW
Potencia consumida nodo emisor 2	1227,7 mW
Potencia consumida nodo emisor 3	627 mW
Potencia consumida dispositivo emisor 1	410 mW
Potencia consumida dispositivo emisor 2	550 mW
Potencia consumida dispositivo emisor 3	627 mW
Potencia consumida emisor HFBR – 1528 sensor pulido Nivel	271,65 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir Nivel	283,3 mW
Potencia consumida fotodetector IF-D91 sensor pulido Nivel	0,165 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir Nivel	0,315 mW

Potencia consumida emisor HFBR – 1528 sensor pulido T ^a	328,4 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir T ^a	348,8 mW
Potencia consumida fotodetector IF-D91 sensor pulido T ^a	0,016 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir T ^a	0,555 mW
Alcance sin paredes (según fabricante)	100 m
Alcance con paredes	17 - 28 m

Tabla 13. Especificaciones de la WSN.

Tras realizar una comparativa con las diferentes tecnologías inalámbricas de comunicación que comparten la banda de 2,4 GHz sin licencia, se ha optado por una comunicación ZigBee porque:

- Brinda la flexibilidad de la conexión de redes en malla.
- Tiene reducidos requisitos de memoria.
- Menor consumo eléctrico y menor tasa de transferencia que bluetooth y WiFi, la Tabla 14 recoge una comparativa entre el consumo de las diferentes tecnologías según sus fabricantes.
- Comunicaciones a través de un único canal.
- Posibilidad de tener hasta 65.535 equipos en la red.

COMPARATIVA CONSUMO PRINCIPALES ESTÁNDARES DE COMUNICACIÓN		
ZigBee	Bluetooth	WiFi
Estándar 802.15.4.	Estándar 802.15.1.	Estándar 802.11.
TX 35 mA	TX 40 Ma	TX > 400 mA
Standby 3μA	Standby200 μA	Standby 20 mA

Tabla 14. Comparativa del consumo de los estándares de comunicación.

Cada nodo cuenta con un dispositivo Arduino-Diecimila que es una plataforma *open-hardware* basada en una sencilla placa con entradas y salidas (I/O), analógicas y digitales y cuenta con los

módulos XBee transceptores de radio frecuencia que trabajan en la banda de 2.4 GHz con protocolo de comunicación 802.15.4.

El módulo XBee presenta una potencia de salida de 1mW, y la sensibilidad del receptor es de -92dBm. Esto le permite operar hasta a unos 100m en espacios abiertos, y hasta 30m en espacios urbanos según el fabricante.

La red puede contar con un número de 65.535 nodos sensores, en este caso, se han incluido tres. Cada nodo sensor cuenta con dos sensores unidos a dos entradas analógicas, pudiendo tener hasta seis sensores por nodo.

Los sensores utilizados en esta red inalámbrica han sido diseñados con fibra óptica de plástico contando con la ventaja de que por ella no circula corriente eléctrica, lo que hace accesible su uso en ambientes peligrosos y de difícil acceso. La fibra óptica se puede utilizar para medir diferentes parámetros como pueden ser tensión, temperatura, presión, nivel de líquido, etc. y se ha demostrado la capacidad de los sensores basados en POFs para medir los niveles de líquidos en ambientes hostiles, inflamables y de difícil acceso.

Para llegar al diseño final de la WSN se ha dividido el trabajo en tres fases de trabajo:

- Fase I: Toma de contacto con los dispositivos Arduino y la comunicación por RF, implementación de una red punto a punto formada por un nodo emisor con seis sensores unidos a él y un nodo receptor, y diseño de la interfaz gráfica de monitorización de datos.
- Fase II: Implementación de una red punto a multipunto con comunicación unidireccional entre equipos, formada por tres nodos emisores con dos sensores unidos a cada nodo y un nodo receptor, y diseño de la interfaz gráfica de monitorización de datos.
- Fase III: Implementación de una red punto a multipunto con comunicación bidireccional entre equipos, formada por tres nodos emisores con dos sensores unidos a cada nodo y un nodo receptor, y diseño de la interfaz gráfica de monitorización de datos. Obtención de la WSN final.

En la Tabla 15 se recogen las características de cada red diseñada en las diferentes fases de trabajo.

CARACTERÍSTICAS PRINCIPALES DE CADA FASE DE TRABAJO			
	FASE I	FASE II	FASE III
Tipo de red	Punto a Punto	Punto a Multipunto	Punto a Multipunto
Tipo de comunicación	Unidireccional	Unidireccional	Bidireccional
Nº kits Arduino en la red	2	4	4
Nº end devices	1	3	3
Nº end devices máx. que soporta la red	1	65.535	65.535
Nº sensores por end device	6	2	2
Control de envío y recepción de datos	NO	SÍ	SÍ
Modo programación módulos	Transparente	API	API
Microprocesador Coordinador	SÍ	SÍ	NO
Disposición JUMPERS de comunicación en Coordinador	XBee	XBee	USB

Tabla 15. Características de las redes diseñadas en cada Fase de trabajo.

A continuación, en la Tabla 16 se pueden ver las diferentes distancias a las que se ha ido comprobando el alcance de la red. En todos los casos hay paredes de por medio con diferentes espesores y en el caso de los 17 metros los muros eran más gruesos. Por lo tanto, se deduce que nuestro sistema tiene un alcance entre 17 – 28 metros con existencia de paredes. El fabricante nos dice que el alcance son unos 30 metros. Estos datos están recogidos de forma más detallada en el capítulo 6 de este documento.

DISTANCIAS DEL NODO EMISOR AL NODO COORDINADOR CON PAREDES DE POR MEDIO	
DISTANCIA AL NODO COORDINADOR	FALLO EN LA TRANSMISIÓN
10 m	NO
14 m	NO
17 m	SÍ
19 m	NO
26 m	NO
28 m	NO
29 m	SÍ

Tabla 16. Alcance de la red con paredes.

La aplicación para monitorizar datos definitiva, es la que aparece en la Figura 45 donde se tienen:

- Seis relojes de medida, dos por equipo emisor, y cada equipo representa cada uno de los sensores del sistema.
- Seis LEDs de color rojo sobre cada reloj que se ilumina cuando se produce una variación en el valor de medida de cada sensor.
- Tres LEDs verdes, uno por equipo, que permanece encendido cuando alguno de los sensores sobrepasa los 5 Voltios establecidos como límite máximo. Se puede decir que hace de alarma.
- Los botones de configuración del puerto serie de comunicación, que abre la pantalla para configurar la configuración, y el botón de salir, que permite abandonar la aplicación.
- Un botón de testeo que brinda al usuario la posibilidad de comprobar si las medidas representadas son las que están siendo medidas en ese momento.

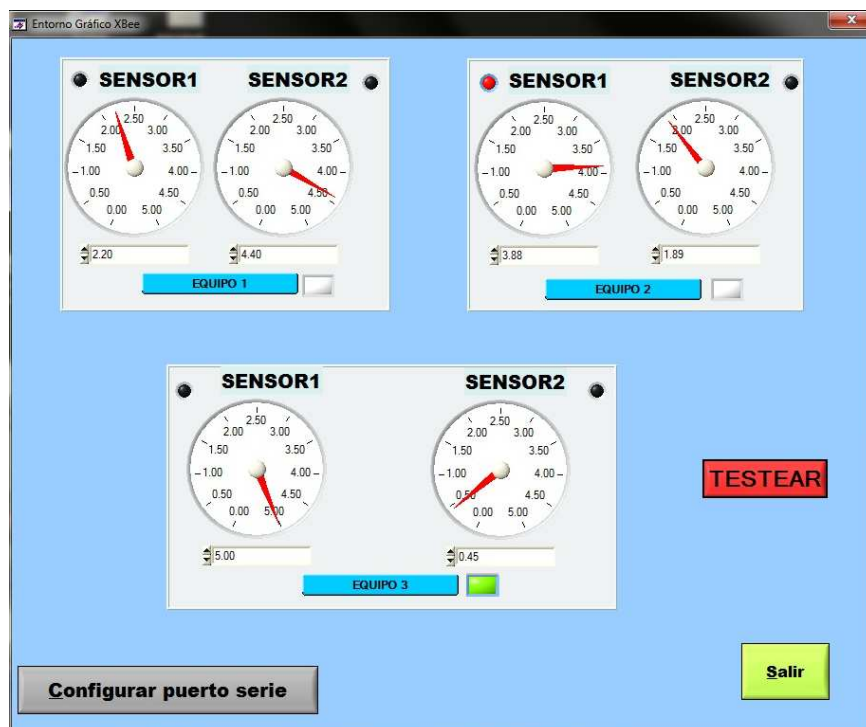



Figura 45. Pantalla perteneciente al diseño final.

- Creación de un fichero .txt donde se recogen todos los datos de medida de nuestro sistema. Dicho fichero se guarda en la carpeta donde se encuentra el ejecutable de la interfaz gráfica. En la Figura 46 aparece el fichero creado, cuyo nombre es data.txt. El archivo cuenta con cinco columnas:
 - Tiempo. Recoge la hora en la que se produce una nueva medida.
 - Equipo. Indica el nodo emisor que envía los datos.
 - Sensor. Salida analógica a la está unido el sensor que ha variado.
 - Lectura. Datos de medida.
 - Alarma. Se pone a '1' cuando el valor monitorizado está por encima de 5V.



TIEMPO	EQUIPO	SENSOR	LECTURA	ALARMA
11:04:51	2	1	3,94	0
11:04:53	3	1	5,00	1
11:04:54	2	1	3,93	0
11:04:55	3	1	0,81	0
11:04:57	3	0	0,33	0
11:03:23	3	0	0,22	0
11:03:24	3	1	0,78	0
11:03:28	3	0	1,73	0
11:03:30	3	0	0,10	0
11:03:32	3	1	1,62	0
11:03:33	3	0	0,09	0
11:03:34	3	1	5,00	1
11:03:39	3	1	5,00	1
11:06:27	3	1	2,55	0
11:06:29	3	1	3,14	0
11:06:31	3	0	2,59	0
11:06:33	3	0	2,57	0
11:06:37	3	1	3,13	0

Figura 46. Fichero data.txt que recoge los datos monitorizados por la aplicación gráfica.

En el caso de la representación de la medida de los sensores, se ha optado por representar la tensión de salida de su circuito acondicionador de forma que se pueda generalizar su aplicación a cualquier sensor que presente una salida que varíe de 0 a 5V y que se pueda convertir a la magnitud medida a través de su correspondiente tabla de calibración.

8.2. Nodo Receptor.

El nodo receptor está formado por un dispositivo Arduino, con su respectivo módulo de RF XBee, que hace de coordinador y un PC donde se desarrolla la aplicación gráfica encargada de gestionar los datos recibidos desde el coordinador. Estos datos son enviados desde los diferentes nodos sensores mediante tramas de datos que contienen:

- El equipo que está enviando la información.
- El canal de lectura. De qué sensor procede la información.
- El valor de la medida realizada por el sensor.

Estos datos son recibidos por RF por el coordinador, y éste se los pasa por USB a la interfaz gráfica, para que ésta monitorice los datos recibidos.

El coordinador tiene la posibilidad de comunicarse con cada nodo emisor no sólo recibiendo datos de ellos, si no también, ser él quien los envía. Con esto se tiene la opción de poder comprobar que los nodos están funcionando correctamente en cualquier momento mediante el botón de Testeo que posee la interfaz gráfica. Esta comunicación en ambos sentidos se consigue gracias a la retirada del microprocesador de la placa Arduino.

Para que un dispositivo Arduino pueda actuar como coordinador debe tener configurado su módulo XBee con los parámetros que aparecen en la Tabla 17 mediante el *software* XCTU.

	COORDINADOR
<i>Destination Address(DH)(DL)</i>	0
<i>Source Address(MY)</i>	0
<i>Coordinator Enable(CE)</i>	1
<i>Api Enable(AP)</i>	1

Tabla 17. Parámetros de configuración módulo XBee como coordinador.

En la Tabla 18 se recogen las especificaciones del nodo receptor incluido en esta red sensorial.

ESPECIFICACIONES DEL NODO RECEPTOR	
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Estándar de comunicación	802.15.4.
Nº kits Arduino utilizados como nodos receptores	1
Nº de nodos receptores	1
Control de envío y recepción de datos	SÍ
Modo programación módulos	API
Microprocesador en el Arduino	NO
Disposición JUMPERS de comunicación en el Arduino	USB
Ancho de Banda	250 kbps
Alimentación módulo Arduino	5V
Potencia consumida nodo coordinador	350 mW
Alcance sin paredes según fabricante	100 m
Alcance con paredes	17 – 28 m

Tabla 18. Especificaciones del nodo receptor.

8.3. Nodos emisores.

La red diseñada cuenta con tres nodos emisores formados por un dispositivo Arduino, con su respectivo módulo de RF XBee cada uno, aunque existe la posibilidad de tener hasta 65.535 dispositivos emisores en la red. Cada nodo se encarga de detectar las variaciones que se produzcan en los sensores que a él van unidos y transmitirle los nuevos valores de medida al coordinador para que éste los monitorice.

Los nodos sensores se han utilizado para medir:

- Variaciones de temperatura.
- Variaciones discretas del nivel de líquido.

Cada nodo tiene unido a él dos sensores de fibra óptica de plástico, pudiendo existir hasta un máximo de seis sensores por nodo, encargados de medir las diferentes variaciones mencionadas anteriormente (un nodo mide temperatura y el otro nivel). Un tercer nodo se ha utilizado para comprobar el alcance de la red ZigBee.

Los *end device* se programan en modo API para poder introducir más módulos en la red y los parámetros necesarios que debe tener configurado el módulo XBee para que un dispositivo Arduino pueda actuar como emisor aparecen en la Tabla 19.

	END DEVICE 1	END DEVICE 2	END DEVICE 3
Destination Address(DH)(DL)	0	0	0
Source Address(MY)	1	2	3
Coordinator Enable(CE)	0	0	0
Api Enable(AP)	1	1	1

Tabla 19. Parámetros de configuración módulos XBee como dispositivos finales.

Cuando se produce una variación en la entrada de datos de alguno de los nodos *end device*, el microprocesador integrado en la placa Arduino realiza una lectura de los datos de entrada y le envía por comunicación serie una trama de datos al módulo XBee para que éste se los envíe por RF al coordinador. En la Tabla 20 se recogen las especificaciones de los nodos emisores de la red sensorial.

ESPECIFICACIONES DE LOS NODOS EMISORES	
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Estándar de comunicación	802.15.4.
Nº kits Arduino utilizados como nodos emisores	3
Nº <i>end devices</i> máx. que soporta la red	65.535
Nº sensores por <i>end device</i> utilizados	2
Nº sensores por <i>end device</i> posibles	6
Control de envío y recepción de datos	SÍ
Modo programación módulos	API
Microprocesador en el Arduino	SÍ
Disposición JUMPERS de comunicación en el Arduino	XBee
Ancho de Banda	250 kbps
Alimentación módulo Arduino	5V
Potencia consumida dispositivo emisor 1	410 mW
Potencia consumida dispositivo emisor 2	550 mW
Potencia consumida dispositivo emisor 3	627 mW
Alcance sin paredes según fabricante	100 m
Alcance con paredes	17 – 28 m

Tabla 20. Especificaciones de los nodos emisores.

8.4. Sensores de Nivel.

Uno de los nodos emisores cuenta con dos sensores de medida de niveles discretos de fibra óptica de plástico. Se ha eliminado de las dos fibras el revestimiento mecánico, teniendo una de ellas también pulida su cubierta, para obtener dos medidas de nivel de una forma más sensible, y se le ha dado una curvatura a cada fibra. A cada uno de los nodos le corresponde una fibra pelada y otra fibra pelada y pulida.

Cada fibra dispone de un sistema de emisión (HFBR-1528) encargado de generar la luz que transporta la fibra óptica al depósito, y que posteriormente será recibida por el fotodiodo IF D91 que convierte la

luz procedente de la fibra óptica en corriente. Esta corriente a su vez, es transformada en tensión mediante una resistencia, siendo ésta la medida monitorizada en la aplicación gráfica.

En la Figura 47 aparece un esquemático sobre las uniones entre el emisor, la fibra óptica y el fotodiodo.

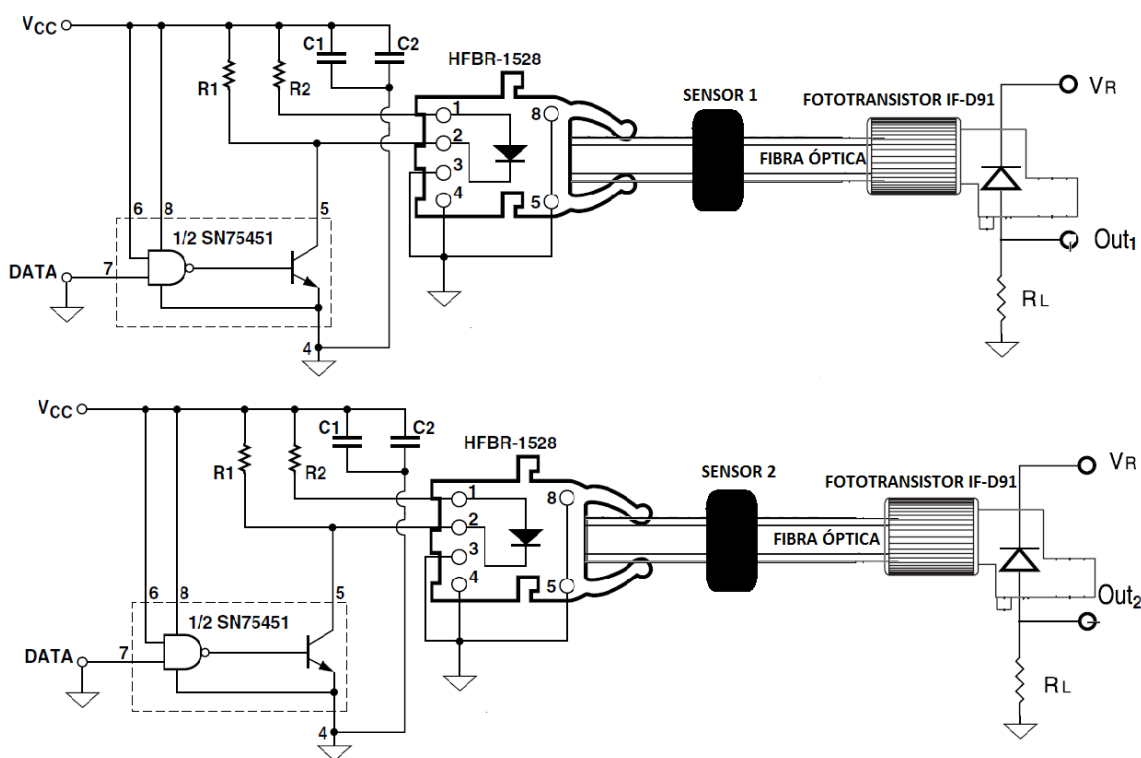


Figura 47. Esquemático sensores de nivel.

El sensado de nivel de líquido se ha llevado a cabo con dos pequeños depósitos, uno de agua y otro de aceite, donde la fibra óptica es sumergida para medir las variaciones que en ellos se producen. En la Figura 48 aparecen estos montajes divididos en cuatro imágenes. Las dos imágenes superiores muestran los dos sensores de fibra óptica, unidos entre sí, y los dos depósitos donde se van a realizar las medidas de nivel. Las dos imágenes inferiores muestran el sistema funcionando con los sensores midiendo en un depósito lleno de agua y otro lleno de aceite.

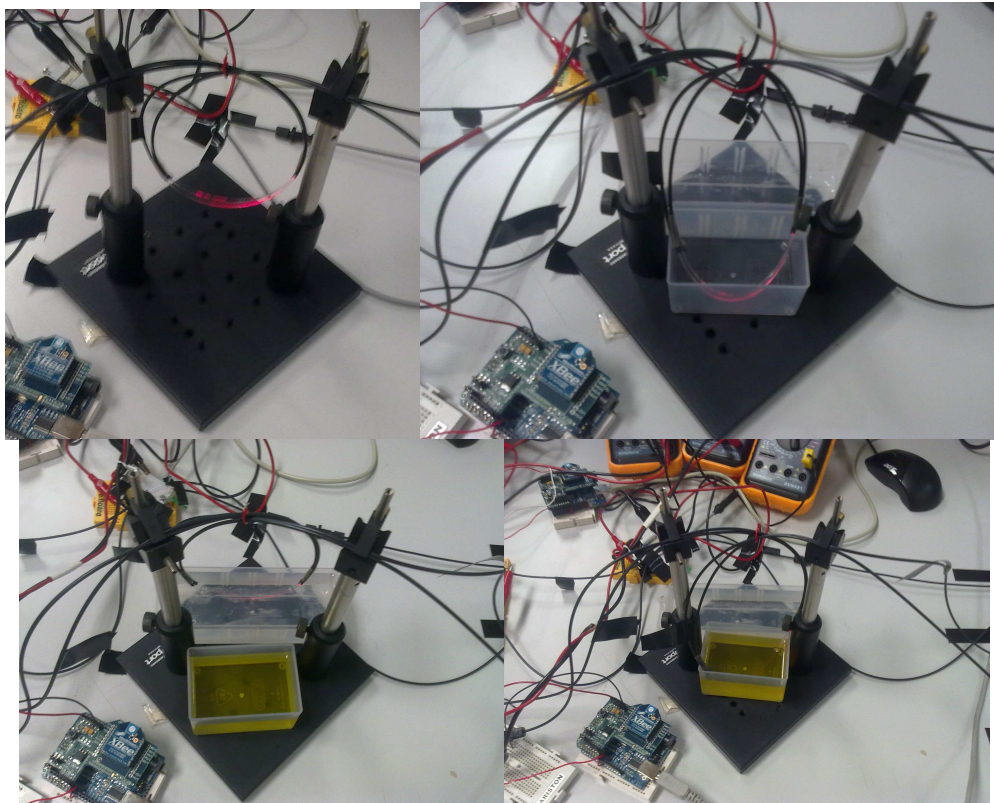


Figura 48. Sensores POF de nivel.

En la Tabla 21 se recogen las especificaciones de estos sensores de nivel.

ESPECIFICACIONES DE LOS SENSORES DE NIVEL	
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Estándar de comunicación	802.15.4.
Nº de emisores HFBR – 1528	2
Nº de fotodetectores IF-D91	2
Nº de fibras utilizadas	2
Nº sensores de nivel por <i>end device</i> utilizados	2
Nº total sensores de nivel en la red	2
Alimentación emisores HFBR – 1528	5 V
Alimentación fotodetector IF-D91	5 V
Potencia consumida emisor HFBR – 1528 sensor pulido Nivel	271,65 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir Nivel	283,3 mW
Potencia consumida fotodetector IF-D91 sensor pulido Nivel	0,165 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir Nivel	0,315 mW
Rango variable de entrada	0-5 V
Rango variable de salida	0-5 V

Tabla 21. Especificaciones de los sensores de nivel.

A continuación, en la Tabla 22 y en la Tabla 23 se muestra la tensión de salida del circuito acondicionador de los sensores de nivel (Vin emisor) y el valor que aparece representado en la interfaz gráfica (Vrepresentada), junto con el valor de la precisión que se produce entre la medida real y la medida representada (diferencia).

La columna Vin emisor se corresponde con la medida tomada a la salida del sensor de nivel, es decir, la que debe ser enviada por el nodo emisor al nodo coordinador una vez se realiza la conversión analógica digital en el dispositivo emisor. La Vrepresentada es la medida que se obtiene a la salida de la conversión analógica digital en el nodo emisor y la enviada en la trama de datos al coordinador. Esta Vrepresentada es la medida representada en el reloj analógico correspondiente al sensor en la aplicación gráfica, y la columna diferencia se corresponde con la precisión que se produce entre la medida real y la representada. Esta diferencia es debida a la conversión analógica-digital en el nodo emisor.

MEDIDAS DE NIVEL DEPÓSITO CON AGUA SENSOR 1 (Sin pulir)							
Estado del depósito	Vin emisor [V]	V representada [V]	Diferencia [V]	I fotod [mA]	P fotod [mW]	I emisor [mA]	P emisor [mW]
Vacio	3,05	3,04	0,01	0,063	0,315	56,66	283,3
Por la mitad	3,05	3,04	0,01	0,062	0,31	56,66	283,3
Lleno	3,04	3,04	0	0,063	0,315	56,66	283,3
MEDIDAS DE NIVEL DEPÓSITO CON AGUA SENSOR 2(pulida)							
Estado del depósito	Vin emisor [V]	V representada [V]	Diferencia [V]	I fotod [mA]	P fotod [mW]	I emisor [mA]	P emisor [mW]
Vacio	2,82	2,82	0	0,033	0,165	54,33	271,65
Por la mitad	2,52	2,52	0	0,031	0,155	54,33	271,65
Lleno	2,42	2,41	0,01	0,030	0,15	54,33	271,65

Tabla 22. Medidas sensores de nivel en el depósito de agua.

MEDIDAS DE NIVEL DEPÓSITO CON ACEITE SENSOR 1(Sin pulir)							
Estado del depósito	Vin emisor [V]	V representada [V]	Diferencia [V]	I fotod [mA]	P fotod [mW]	I emisor [mA]	P emisor [mW]
Vacio	2,99	2,99	0	0,061	0,305	56,66	283,3
Por la mitad	2,77	2,76	0,01	0,057	0,285	56,66	283,3
Lleno	2,76	2,76	0	0,057	0,285	56,66	283,3
MEDIDAS DE NIVEL DEPÓSITO CON ACEITE SENSOR 2(pulida)							
Estado del depósito	Vin emisor [V]	V representada [V]	Diferencia [V]	I fotod [mA]	P fotod [mW]	I emisor [mA]	P emisor [mW]
Vacio	2,35	2,35	0	0,028	0,14	54,33	271,65
Por la mitad	1,97	1,97	0	0,023	0,115	54,33	271,65
Lleno	1,94	1,93	0,01	0,023	0,115	54,33	271,65

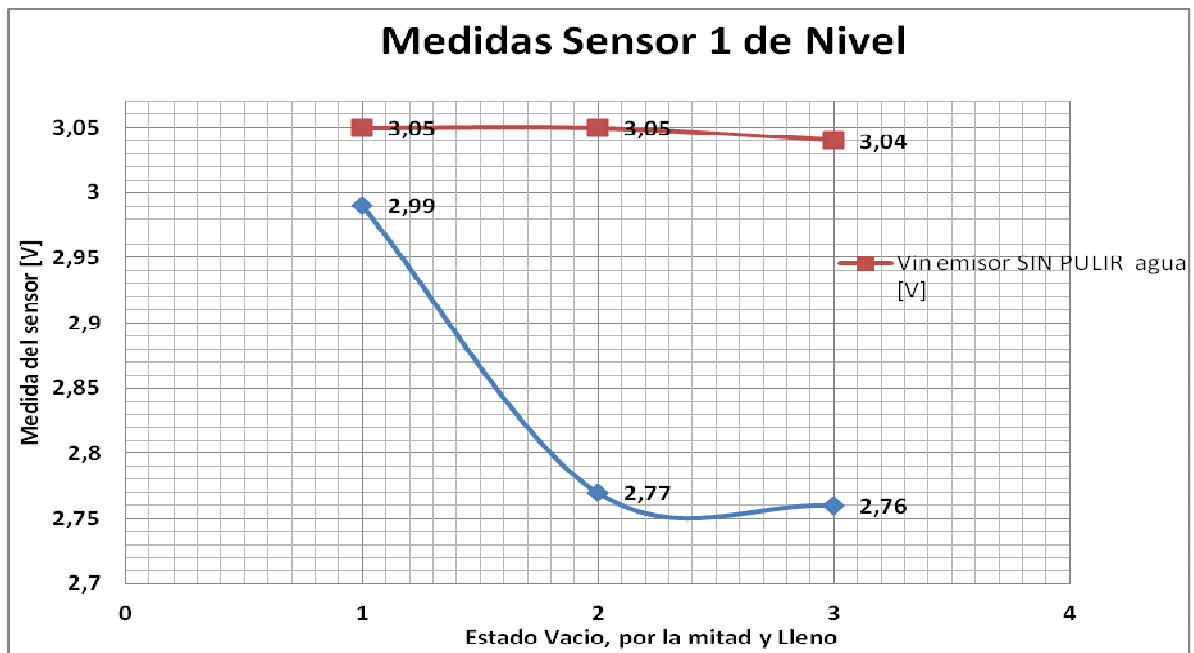
Tabla 23. Medidas sensores de nivel en el depósito de aceite.

Debido a las dos posibilidades de alimentación de los módulos Arduino, mediante puerto USB o con un cargador de red, y a las pequeñas variaciones en la tensión suministrada entre diferentes USB, es necesario realizar una calibración de los dispositivos Arduinos en función de dónde van a ir conectados como se ha explicado en el capítulo 7.

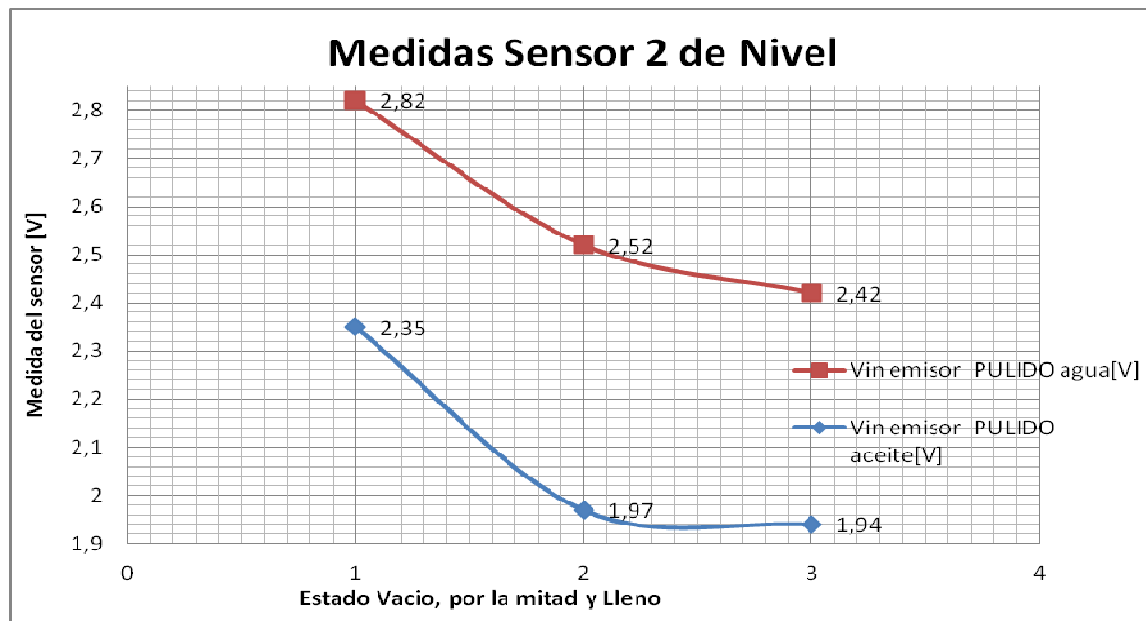
Esta calibración es necesaria por la conversión analógica-digital de 10 bits que se produce al realizar una lectura de los datos de entrada de cada sensor.

La aplicación gráfica diseñada muestra dos decimales, por lo que para un mismo valor de representación se van a tener varios resultados a la salida del ADC, ya que su resolución es de 1LSB = 4 mV, de ahí la diferencia en algunos casos de 0,01 V entre el valor de entrada al módulo emisor y el valor de la medida representada.

A continuación, en la Gráfica 1 y en la Gráfica 2 se recogen las medidas de la tensión de salida de cada sensor de nivel, correspondiente con la tensión de entrada de cada nodo emisor, en función del estado del depósito: vacío, por la mitad, y lleno.



Gráfica 1. Representación de las medidas de los sensores de nivel de POF sin pulir.



Gráfica 2. Representación de las medidas de los sensores de nivel de POF pulida.

8.5. Sensores de Temperatura.

Uno de los nodos emisores cuenta con dos sensores de temperatura de fibra óptica de plástico. Se ha eliminado de las dos fibras el revestimiento mecánico, en una de ellas se ha pulido su cubierta, para obtener dos medidas de temperatura de una forma más sensible. A cada uno de los nodos le corresponde una fibra pelada y otra fibra pelada y pulida.

Como en el caso de los sensores de nivel, cada fibra dispone de un sistema de emisión (HFBR-1528) encargado de generar la luz que transporta la fibra óptica, y que posteriormente será recibida por el fotodiodo IF D91 que convierte la luz procedente de la fibra óptica en corriente. Esta corriente a su vez, es transformada en tensión mediante una resistencia, siendo esta la medida monitorizada en la aplicación gráfica.

En la Figura 49 aparece un esquemático sobre las uniones entre el emisor, la fibra óptica y el fotodiodo.

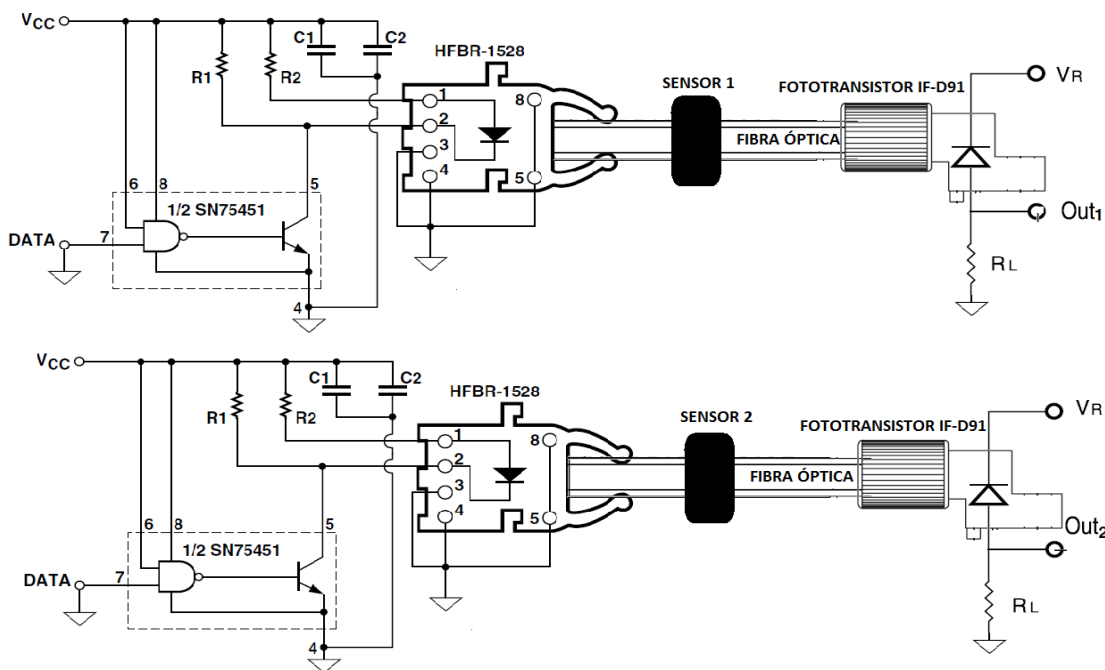


Figura 49. Esquemático sensores de temperatura.

La temperatura se ha medido uniendo la fibra óptica a una placa calefactora utilizada como foco de calor, como muestra la Figura 50. Se ha utilizado una placa calefactora LTS350 para medida de temperatura en cristal líquido como foco de calor, que provoca un aumento de temperatura como consecuencia de las variaciones de temperatura introducidas en la Unidad de control TP94, teniendo así diferentes medidas en función de las variaciones de temperatura producidas.

El principio de funcionamiento del sistema ha sido explicado en el capítulo 5 de este documento y se puede encontrar de forma más detallada en [Tap+11].

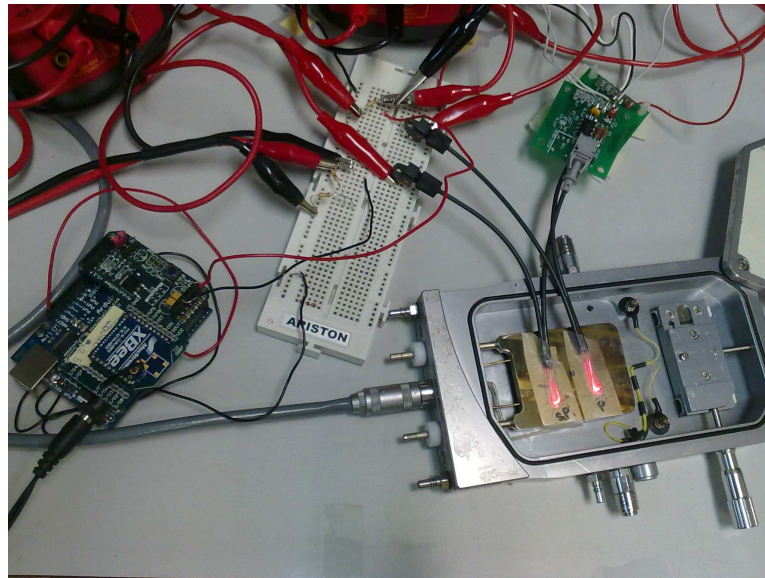


Figura 50. Sensores POF de temperatura unidos a la placa calefactora.

En la Figura 51 aparece la unidad de control de la placa calefactora. La temperatura de la placa se incrementará los grados que marca el display de esta placa una vez se seleccionan los grados deseados.

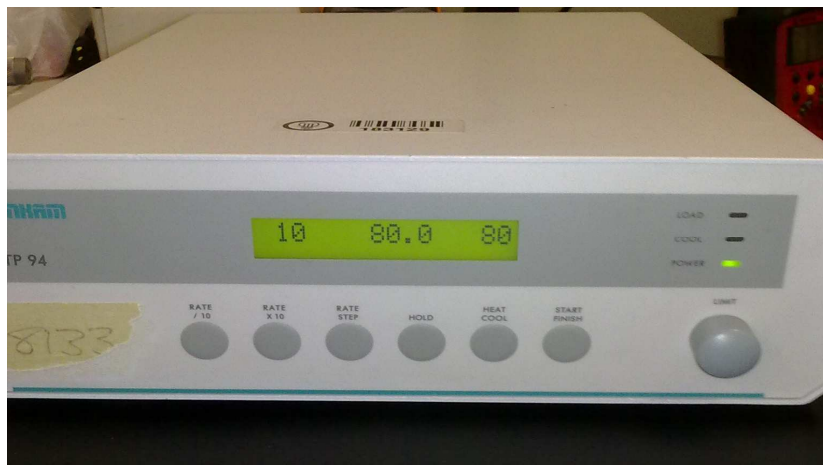


Figura 51. Unidad de control de la placa calefactora.

En la Tabla 24 se recogen las especificaciones de estos sensores de temperatura.

ESPECIFICACIONES DE LOS SENSORES DE TEMPERATURA	
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Estándar de comunicación	802.15.4.
Nº de emisores HFBR – 1528	2
Nº de fotodetectores IF-D91	2
Nº de fibras utilizadas	2
Nº sensores de Tª por <i>end device</i> utilizados	2
Nº total sensores de Tª en la red	2
Placa calefactora LTS350	1
Unidad de control TP94	1
Alimentación emisores HFBR – 1528	5 V
Alimentación fotodetector IF-D91	5 V
Potencia consumida emisor HFBR – 1528 sensor pulido Tª	328,4 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir Tª	348,8 mW
Potencia consumida fotodetector IF-D91 sensor pulido Tª	0,016 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir Tª	0,555 mW
Rango variable de entrada	0-5 V
Rango variable de salida	0-5 V

Tabla 24. Especificaciones de los sensores de temperatura.



A continuación se muestra en la Tabla 25 y en la Tabla 26 la tensión de salida del circuito acondicionador de los sensores de temperatura ($V_{in\text{ emisor}}$) y el valor que aparece representado en la interfaz gráfica ($V_{representada}$), junto con el valor de la precisión que se produce entre la medida real y la medida representada (diferencia).

Como se ha explicado anteriormente, la columna $V_{in\text{ emisor}}$ se corresponde con la medida tomada a la salida del sensor de T^a , es decir, la que debe ser enviada por el nodo emisor al nodo coordinador una vez se realiza la conversión analógica digital en el dispositivo emisor. La $V_{representada}$ es la medida que se obtiene a la salida de la conversión analógica digital en el nodo emisor y la enviada en la trama de datos al coordinador. Esta $V_{representada}$ es la medida representada en el reloj analógico correspondiente al sensor en la aplicación gráfica, y la columna diferencia se corresponde con la precisión que se produce entre la medida real y la representada. Esta diferencia es debida a la conversión analógica-digital en el nodo emisor.

Para la realización de la calibración de los sensores de T^a se han ido tomando medidas cada minuto, momento en el que se incrementa la T^a 2 °C.

MEDIDAS DE TEMPERATURA SENSOR 1 SIN PULIR								
t [min]	T ^a [°C]	Vin emisor [V]	V representada [V]	Diferencia [V]	I fotod [mA]	P fotod [mW]	I emisor [mA]	P emisor [mW]
0	30	2,11	2,11	0	0,107	0,535	69,76	348,8
1	32	2,11	2,11	0	0,1072	0,536	69,76	348,8
2	34	2,12	2,11	0,01	0,1073	0,5365	69,76	348,8
3	36	2,12	2,12	0	0,1055	0,5275	69,76	348,8
4	38	2,12	2,12	0	0,1077	0,5385	69,76	348,8
5	40	2,13	2,12	0,01	0,1078	0,539	69,76	348,8
6	42	2,13	2,13	0	0,108	0,54	69,76	348,8
7	44	2,13	2,13	0	0,1081	0,5405	69,76	348,8
8	46	2,14	2,13	0,01	0,1083	0,5415	69,76	348,8
9	48	2,14	2,13	0,01	0,1084	0,542	69,76	348,8
10	50	2,14	2,14	0	0,1086	0,543	69,76	348,8
11	52	2,15	2,14	0,01	0,1088	0,544	69,76	348,8
12	54	2,15	2,14	0,01	0,1089	0,5445	69,76	348,8
13	56	2,15	2,14	0,01	0,109	0,545	69,76	348,8
14	58	2,15	2,16	0,01	0,1092	0,546	69,76	348,8
15	60	2,16	2,16	0	0,1093	0,5465	69,76	348,8
16	62	2,16	2,16	0	0,1095	0,5475	69,76	348,8
17	64	2,16	2,16	0	0,1096	0,548	69,76	348,8
18	66	2,16	2,16	0	0,1097	0,5485	69,76	348,8
19	68	2,17	2,17	0	0,1099	0,5495	69,76	348,8
20	70	2,17	2,17	0	0,11	0,55	69,76	348,8
21	72	2,17	2,17	0	0,1103	0,5515	69,76	348,8
22	74	2,18	2,18	0	0,1104	0,552	69,76	348,8
23	76	2,18	2,18	0	0,1105	0,5525	69,76	348,8
24	78	2,18	2,18	0	0,1107	0,5535	69,76	348,8
25	80	2,19	2,18	0,01	0,1109	0,5545	69,76	348,8

Tabla 25. Medidas sensor de temperatura POF sin pulir.

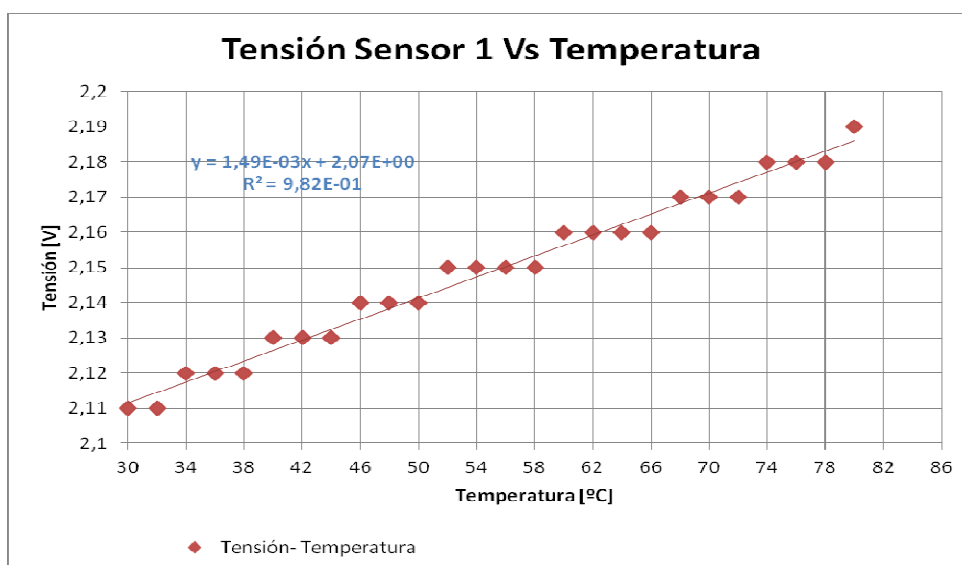
MEDIDAS DE TEMPERATURA SENSOR 2 PULIDO								
t [min]	T ^a [°C]	Vin emisor [V]	V representada [V]	Diferencia [V]	I fotod [mA]	P fotod [mW]	I emisor [mA]	P emisor [mW]
0	30	2,77	2,76	0,01	0,00305	0,01525	69,76	348,8
1	32	2,78	2,77	0,01	0,00306	0,0153	69,76	348,8
2	34	2,79	2,78	0,01	0,00307	0,01535	69,76	348,8
3	36	2,80	2,78	0,02	0,00308	0,0154	69,76	348,8
4	38	2,81	2,80	0,01	0,00309	0,01545	69,76	348,8
5	40	2,82	2,81	0,01	0,0031	0,0155	69,76	348,8
6	42	2,83	2,82	0,01	0,00311	0,01555	69,76	348,8
7	44	2,84	2,83	0,01	0,00312	0,0156	69,76	348,8
8	46	2,85	2,84	0,01	0,00314	0,0157	69,76	348,8
9	48	2,86	2,86	0	0,00315	0,01575	69,76	348,8
10	50	2,87	2,87	0	0,00316	0,0158	69,76	348,8
11	52	2,88	2,87	0,01	0,00316	0,0158	69,76	348,8
12	54	2,89	2,88	0,01	0,00317	0,01585	69,76	348,8
13	56	2,90	2,89	0,01	0,00319	0,01595	69,76	348,8
14	58	2,91	2,90	0,01	0,00319	0,01595	69,76	348,8
15	60	2,92	2,91	0,01	0,0032	0,016	69,76	348,8
16	62	2,93	2,92	0,01	0,00321	0,01605	69,76	348,8
17	64	2,93	2,92	0,01	0,00322	0,0161	69,76	348,8
18	66	2,94	2,94	0	0,00323	0,01615	69,76	348,8
19	68	2,96	2,96	0	0,00325	0,01625	69,76	348,8
20	70	2,95	2,94	0,01	0,00324	0,0162	69,76	348,8
21	72	2,96	2,95	0,01	0,00325	0,01625	69,76	348,8
22	74	2,98	2,97	0,01	0,00326	0,0163	69,76	348,8
23	76	2,98	2,98	0	0,00327	0,01635	69,76	348,8
24	78	2,99	2,98	0,01	0,00328	0,0164	69,76	348,8
25	80	3,00	2,99	0,01	0,00329	0,01645	69,76	348,8

Tabla 26. Medidas sensor de temperatura POF pulido.

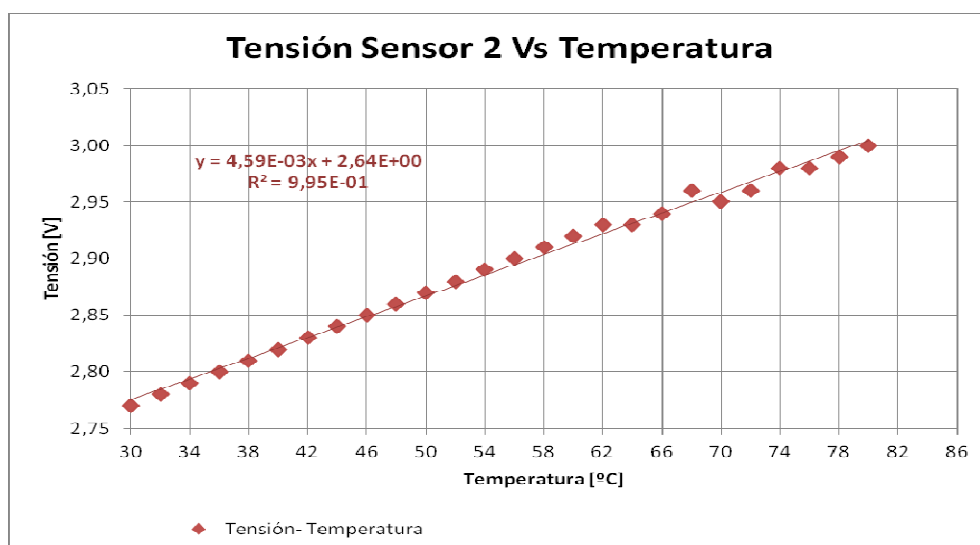
De nuevo, hay que tener en cuenta lo explicado en el caso de los sensores de medida con respecto a la diferencia que se produce entre la Vin emisor y la Vin representada. La aplicación gráfica diseñada muestra dos decimales, por lo que para un mismo valor de representación se van a tener varios

resultados a la salida del ADC, ya que su resolución es de 1LSB = 4 mV, de ahí la diferencia en algunos casos de 0,01 V entre el valor de entrada al módulo emisor y el valor de la medida representada.

A continuación, en la Gráfica 3 y la Gráfica 4 se representan los valores de la tensión de salida de los sensores de POF frente a la variación de temperatura. Tensión y T^a se pueden aproximar mediante una ecuación lineal como se observa en cada una de las gráficas, y con su ecuación se podría representar la variación de T^a en el sensor en la aplicación gráfica.



Gráfica 3. Representación de las medidas de los sensores de T^a de POF sin pulir.



Gráfica 4. Representación de las medidas de los sensores de T^a de POF pulida.



Capítulo 9. Conclusiones y Trabajos futuros.

Los objetivos iniciales al comienzo de este proyecto eran:

- Trabajar en un medio inalámbrico que proporcionara movilidad y flexibilidad al sistema de medida.
- Uso de técnicas ópticas que permitan la medida segura en entornos difíciles (riesgo explosión, exceso de radiación, biocompatibles, etc.)
- Controlar los sensores de fibra óptica con dispositivos inalámbricos.

- Sistema de bajo consumo.
- Diseñar un entorno gráfico amigable de fácil y sencillo manejo a la vez que proporcione una rapidez de lectura de los datos recogidos.

Tras la consecución de este proyecto se puede decir que:

- Se ha diseñado un sistema de control sensorial por radiofrecuencia mediante comunicación ZigBee con las especificaciones que aparecen en la Tabla 27.

ESPECIFICACIONES DE LA RED SE SENSORES INALÁMBRICA	
Tipo de red	Punto a Multipunto
Tipo de comunicación	Bidireccional
Estándar de comunicación	802.15.4.
Nº kits Arduino en la red	4
Nº <i>end devices</i>	3
Nº <i>end devices</i> máx. que soporta la red	65.535
Nº sensores por <i>end device</i> utilizados	2
Nº sensores por <i>end device</i> posibles	6
Nº de emisores HFBR – 1528	4
Nº de fotodetectores IF-D91	4
Nº de fibras utilizadas	4
Nº sensores de nivel por <i>end device</i> utilizados	2
Nº total sensores de nivel en la red	2
Nº sensores de Tª por <i>end device</i> utilizados	2
Nº total sensores de Tª en la red	2
Placa calefactora LTS350	1
Unidad de control TP94	1
Alimentación emisores HFBR – 1528	5 V
Alimentación fotodetector IF-D91	5 V
Modo programación módulos	API
Microprocesador Coordinador	NO
Disposición JUMPERS de comunicación en Coordinador	USB
Ancho de Banda	250 kbps
Alimentación módulos Arduino	5 V
Potencia consumida nodo receptor	350 mW
Potencia consumida nodo emisor 1	965,43 mW
Potencia consumida nodo emisor 2	1227,7 mW
Potencia consumida nodo emisor 3	627 mW
Potencia consumida dispositivo emisor 1	410 mW
Potencia consumida dispositivo emisor 2	550 mW

Potencia consumida dispositivo emisor 3	627 mW
Potencia consumida emisor HFBR – 1528 sensor pulido Nivel	271,65 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir Nivel	283,3 mW
Potencia consumida fotodetector IF-D91 sensor pulido Nivel	0,165 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir Nivel	0,315 mW
Potencia consumida emisor HFBR – 1528 sensor pulido T^a	328,4 mW
Potencia consumida emisor HFBR – 1528 sensor sin pulir T^a	348,8 mW
Potencia consumida fotodetector IF-D91 sensor pulido T^a	0,016 mW
Potencia consumida fotodetector IF-D91 sensor sin pulir T^a	0,555 mW
Alcance sin paredes (según fabricante)	100 m
Alcance con paredes	17 - 28 m

Tabla 27. Especificaciones de la Red Sensorial Inalámbrica.

- El sistema diseñado es capaz de realizar la lectura de dos entradas analógicas (sensores con fibras ópticas), y detectar la variación de cada una de estas entradas sin necesidad de cableado adicional.
- Los módulos utilizados, Arduino Diecimila + XBee, dan la posibilidad de incorporar un máximo de 6 entradas analógicas.
- El uso de módulos XBee permite un ahorro de energía debido a su bajo consumo y un alcance suficiente a lo requerido en este proyecto.
- La escalabilidad del sistema hace posible incrementar el área monitorizada simplemente añadiendo nodos y sin necesidad de configuración alguna, ya que serán incorporados a la red dinámicamente. Por otra parte, el hecho de que uno de ellos desaparezca no perjudica el funcionamiento global de la red ya que la información puede seguir siendo transmitida por caminos alternativos. Cada nuevo módulo XBee (perteneciente a cada nuevo nodo emisor) debe ser programado mediante el software XCTU habilitándose como dispositivo final, con su dirección, la dirección del coordinador y el modo API. Como en la trama de datos enviada por el emisor al coordinador incluye el identificador del equipo que le envía los datos, el módulo coordinador no necesita ningún cambio de programación.
- Se ha realizado una interfaz gráfica que facilita la visualización de los datos analizados, formado por un ejecutable instalado directamente en el PC y un fichero de texto que recoge los datos monitorizados.



- La diferencia producida entre la medida representada en la aplicación gráfica y la medida real a la salida del sensor, se debe a que la aplicación muestra dos decimales de media, por lo que para un mismo valor de representación se van a tener varios resultados a la salida del conversor analógico-digital de 10 bits, ya que su resolución es de $1\text{LSB} = 4\text{ mV}$.
- Finalmente se ha incorporado una comunicación bidireccional entre cada equipo emisor y el coordinador dando lugar a un sistema:
 - Más fiable.
 - Más flexible.
 - Más robusto.
 - Más visual.

Una vez finalizados el diseño e implementación del sistema, y comprobado su correcto funcionamiento, se puede afirmar que se han cumplido satisfactoriamente los objetivos marcados y se ha conseguido la implantación de una red sensorial donde existe la posibilidad de ampliación en cualquier momento dando flexibilidad al proyecto.

A continuación se detallan los requisitos necesarios para el montaje del sistema diseñado en este proyecto:

- Un Ordenador Intel (R) Core™ i3 CPU 5502 @ 3,20 GHz (Nº inventario 197356).
- Cuatro Kits compuestos de un Arduino Diecimila y módulos de comunicación XBee que permiten la transmisión inalámbrica.
- Cuatro emisores ópticos HFBR- 1528.
- Cuatro fotodetectores IF- D91.



- Fibra óptica de plástico.
- Placa calefactora LTS350.
- Unidad de control TP94.
- Tres Cables USB A-B y un cargador de red.
- Osciloscopio TEKTRONIX, modelo TDS 210.
- Tres fuentes de alimentación AC/DC de ~220 a 9Vcc.
- Mesa óptica TMC, modelo 5085326330 y soportes ópticos.
- Polímetro digital MASTECH, modelo MY-63.
- Software Entorno de Desarrollo de ARDUINO.
- Software XCTU.
- Software LabWindows de National Instruments.

El PC al que va unido el nodo coordinador puede estar ubicado en el lugar que se desee, siempre y cuando su distancia con los nodos emisores cumpla con el alcance de la red especificado: entre 17 y 28 metros con existencia de paredes.

A partir del diseño creado, se pueden realizar trabajos futuros para mejorar el dispositivo una vez se realicen los ensayos a los que está destinado y se compruebe la eficiencia del mismo como por ejemplo:

- Posible ampliación de la red con más módulos Arduino-XBee.
- Introducción de diferentes elementos sensores.
- Realización de pruebas con los nuevos sensores específicos para ajustar más los márgenes de medida.



- Incorporar baterías a los módulos emisores.
- Permitir la posibilidad de que el usuario interactúe más con la aplicación gráfica.

Añadir, que en el Proceedings 20th International Conference on Plastic Optical Fibers Conference, celebrado en Bilbao en 2011 se ha publicado el artículo: **“WIRELESS MESH NETWORK APPLIED TO POLYMER OPTICAL FIBRE-BASED SENSORS”**.D.S. Montero, M. Chaparro de la Peña, C. Vázquez , J. Zubia, haciendo referencia a los estudios realizados en este proyecto.



Capítulo 10. Bibliografía.

[Aak+06] Niels Aakvaag, Jan-Erik Frey (2/2006): “Nuevas soluciones de interconexión para la automatización industrial”. Revista ABB Pág 39-32.

<[http://library.abb.com/GLOBAL/SCOT/scot271.nsf/VerityDisplay/A019E9833DCF2819C1257199004E5DD2/\\$File/39-42%202M631_SPA72dpi.pdf](http://library.abb.com/GLOBAL/SCOT/scot271.nsf/VerityDisplay/A019E9833DCF2819C1257199004E5DD2/$File/39-42%202M631_SPA72dpi.pdf) >

[Arduino] Arduino. Disponible [Internet].

< <http://www.Arduino.cc> >

[Blogelectronica] Blog personal sobre alta tecnología y dispositivos electrónicos avanzados. Disponible [Internet].

< <http://www.blogelectronica.com/zigbee-maxstream-modulos-practica-2/> >



[Bluetooth] The Official Bluetooth Technology Web Site. Disponible [Internet].

< <http://www.Bluetooth.com> >

[Cas+10] Francisco Casellas, Guillermo Velasco, Francesc Guinjoan y Robert Piqué (2010) : “El concepto de Smart Metering en el nuevo escenario de distribución eléctrica”. Seminario Anual de Automática, Electrónica Industrial e Instrumentación. "XVII Seminario Anual de Automática, Electrónica Industrial e Instrumentación". Bilbao: 2010, p. 752-757.

< <http://upcommons.upc.edu/e-prints/bitstream/2117/9066/1/5025.pdf> >

[Datasheet] Datasheet and application notes. Disponible [Internet].

< <http://www.datasheetarchive.com/> >

[Digi] Digi making wireless M2M easy. Disponible [Internet].

< <http://www.digi.com> >

[Dimap] Digital Image Processing S. L. Disponible [Internet].

< <http://www.dimap.es> >

[Educastur] Comunidad educativa de blogs. Disponible [Internet].

< <http://blog.educastur.es/electronica/files/2008/05/manual-labwindows-castellano.pdf> >



[Enerzuul] Enerzuul, 2010 - Crear, ver, analizar y ahorrar energía. Disponible [Internet].

< <http://www.enerzuul.es/?p=781> >

[Gom05] Rubén Gómez Camuñas (2005): “Control remoto de una ventana inteligente basada en PDLC”. Proyecto Fin de Carrera. Universidad Carlos III de Madrid. Ingeniería de Telecomunicación.

[Gas10] David Gascón (2010): “Conjunto de retos y líneas de investigación prioritarias en Redes Sensoriales Inalámbricas para los próximos años”. “. Publicado en la Plataforma Tecnológica Española de comunicaciones inalámbricas.

< http://atc.ugr.es/~aprieto/TIC_socio_sanitario/A11_4_05_Redес_sensores.pdf >

[Gra07] Gralla, Preston (2007). “*Cómo funcionan las redes inalámbricas*”. Anaya Multimedia.

<http://www.idi.aetic.es/eMOV/CLI_AETIC/ftpportalweb/documentos/6_eMOV_AEI_RedSens_web.pdf >

[Libelium] Libelium Comunicaciones Distribuidas S.L. Disponible [Internet].

< <http://www.libelium.com/applications/environment> >

[Lop+10] Gregorio López, Víctor Custodio, and José Ignacio Moreno, Member IEEE (2010): “LOBIN: E-Textile and Wireless-Sensor-Network-Based Platform for Healthcare Monitoring in Future Hospital Environments”. Publicado en IEEE Transactions on Information Technology in Biomedicine pág. 1446 - 1458

< http://e-archivo.uc3m.es/bitstream/10016/12136/1/lobin_IEEE_2010_ps.pdf >



[Manual-Wifi] Redes inalámbricas WiFi, manuales, videos y tutoriales. Disponible [Internet].

< <http://www.manual-wifi.com/tipos-de-redes-inalambricas/> >

[Mon+09] D. S. Montero, C. Vázquez, I. Möllers, J. Arrue, D. Jäeger "A Self-referencing Intensity Based Polymer Optical Fiber for Liquid Detection Sensors" **Sensors** (Special Issue State-of-the-Art Sensors Technology in Spain), 9 (8), 6446-6455, 2009 .

[Mon+10] D.S. Montero, C. Vázquez, P. C. Lallana, J. Zubia "Polymer Optical Fibre Sensor for Liquid Detection Applied to Volumetric Flasks" **19 th Internacional Conference on Plastic Optical Fibers ICPOF 2010** , 2010.Yokohama, Japan

[Pul+10] José Antonio Pullés Boudet, Alfredo Pino Escalona(2010) : “Redes Inalámbricas para el Monitoreo y Control en tiempo real de Sistemas de Calentamiento de agua con Colectores Solares”. Publicado en la revista de Ingeniería Electrónica, Automática y Comunicaciones, Vol 32, No 1, pág. 38-47.

< <http://rielac.cujae.edu.cu/index.php/rieac/article/view/75> >

[Rei+] Esteban Santiago Reinoso Pérez, Ing., María Soledad Jiménez Jiménez : “Diseño e implementación de un prototipo para una red de domótica y seguridad para un hogar utilizando el estándar IEEE 802.15.4”. Presentado en XXII JORNADAS EN INGENIERÍA ELÉCTRICA Y ELECTRÓNICA.

<<http://bibdigital.epn.edu.ec/bitstream/15000/9936/1/DISE%3FO%20E%20IMPLEMENTACI%3FN%20DE%20UN%20PROTOTIPO%20PARA%20UNA%20RED%20DE%20DOM%3FTICA%20Y%20SEGURIDAD%20PARA%20UN%20HOGAR%20UTILIZ.pdf> >



[Ser07] Jesús Serna Sanchis (2007): “Trabajo de ampliación de redes. Redes de sensores inalámbricas”. Universidad de Valencia.

< <http://www.uv.es/montanan/ampliacion/trabajos/Redes%20de%20Sensores.pdf> >

[Tap+11] A. Tapetado, C. Vázquez, J. Zubia “*Temperature sensor based on Polymer Optical Fiber macro-bends*” **20 th Internacional Conference on Plastic Optical Fibers ICPOF 2011**, 2011. Bilbao (Spain)

[Wi-fi] Wi-fi Alliance. Disponible [Internet].

< <http://www.wi-fi.org> >

[ZigBee] ZigBee Alliance. Disponible [Internet].

< <http://www.zigbee.org> >

[ZigBeeexpo] ZigBee Expo 2008. Disponible [Internet].

< <http://www.zigbeeexpo.com/2008/na/> >



Anexo 1. LabWindows. Interfaz gráfica.

LabWindows/CVI es un entorno de desarrollo integrado para programar en lenguaje C. Se basa en un entorno interactivo para el desarrollo de programas y unas librerías de funciones para crear aplicaciones de adquisición de datos y control de instrumentos.

Las herramientas con las que se puede trabajar son:

- Un editor de interfaces gráficas de usuario.
- Una ventana para editar el código fuente de programas ANSI C.
- Paneles de funciones para la ejecución interactiva y generación de código de forma automática.
- Compilador y Depurador integrados para el desarrollo y mantenimiento de proyectos.

La clave de la potencia de LabWindows/CVI está en sus librerías que proporcionan funciones para todas las fases del desarrollo de una aplicación de adquisición de datos o de control de instrumentación. [Educastur]

A continuación se puede ver de forma más detallada como crear un proyecto en LabWindows:

- **Crear nuevo proyecto.** En la barra de herramientas se entra en *File* → *New* → *Project* como indica la Figura 52. Se le da un nombre al proyecto que se quiere crear y se selecciona la opción *Create Project in New Workspace* como muestra la Figura 53.

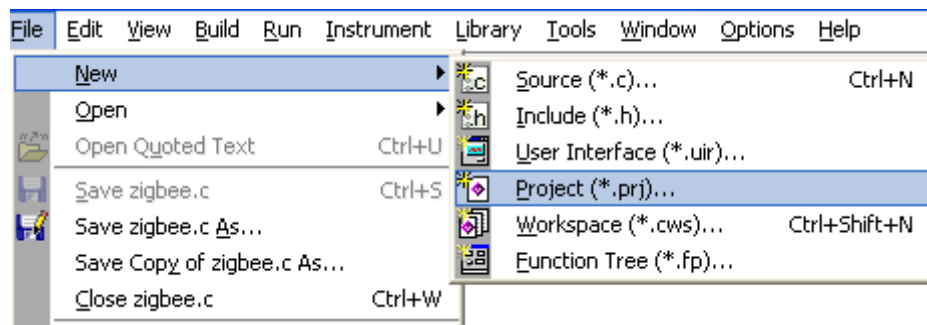


Figura 52. Creación de un nuevo proyecto en LabWindows.

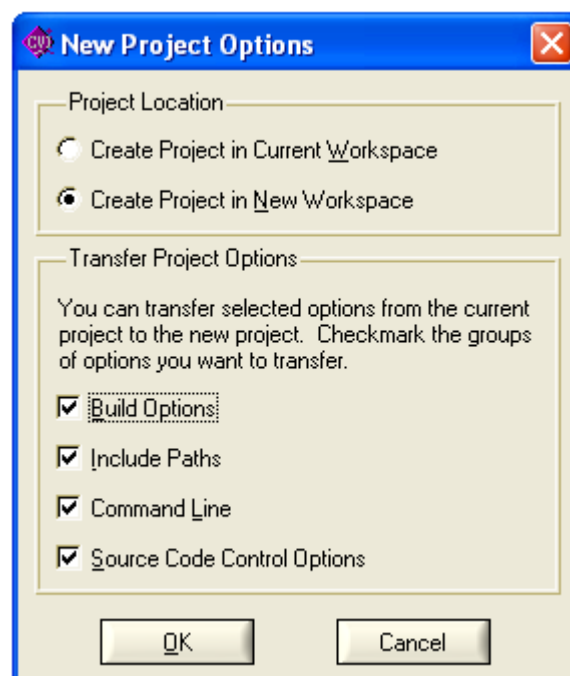


Figura 53. Creación de un workspace en LabWindows.

- **Crear ficheros .c y .uir.** En el fichero .c se encuentran las líneas de código del programa que se va a utilizar. La forma de crearlo es mediante la opción *File → New → Source*.c* como se puede ver en la Figura 54. Una vez creado el fichero el siguiente paso es escribir el código en esta pantalla.

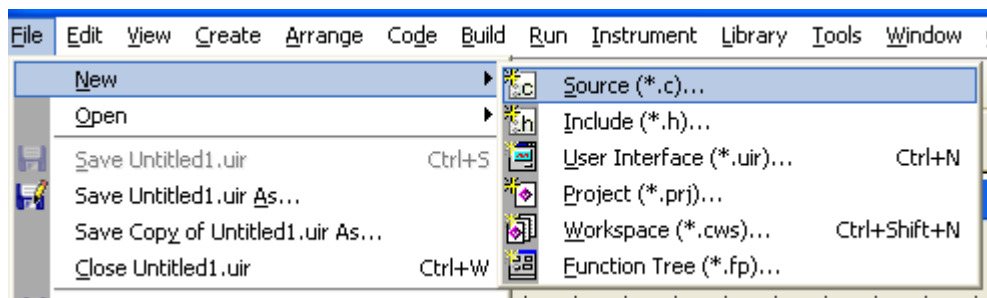


Figura 54. Nuevo fichero .c.

Otro fichero necesario es el .uir, que será donde se edite la interfaz gráfica sin necesidad de generar líneas de código gracias al editor de interfaces de LabWindows. Para crear este fichero se siguen los mismos pasos que aparecen en la Figura 55.

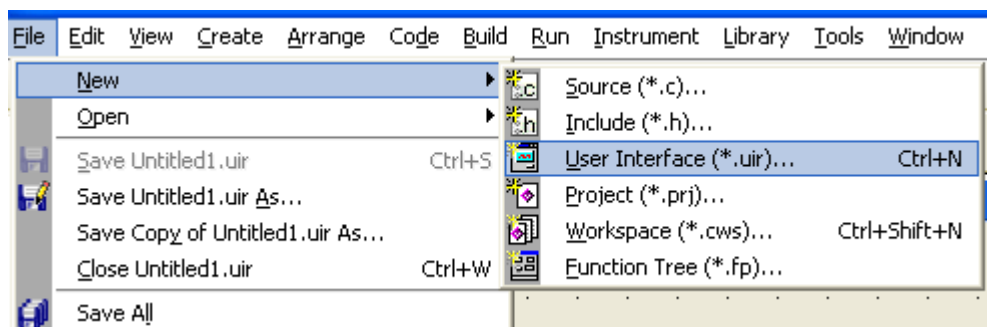


Figura 55. Nuevo fichero .uir.

- **Añadir los ficheros .c, .h y .uir (interfaz gráfica).** A continuación se añaden los distintos ficheros necesarios al proyecto creado para empezar a trabajar en la interfaz gráfica. Desde la pestaña *Edit* → *Add file to project* de la Figura 56, se van añadiendo los ficheros *.c que contendrá el código fuente del programa, las librerías *.h necesarias, en nuestro caso ASCII.h y panel.h y el panel.uir donde se recogen todas las pantallas que se quiere que aparezcan en el diseño realizado.

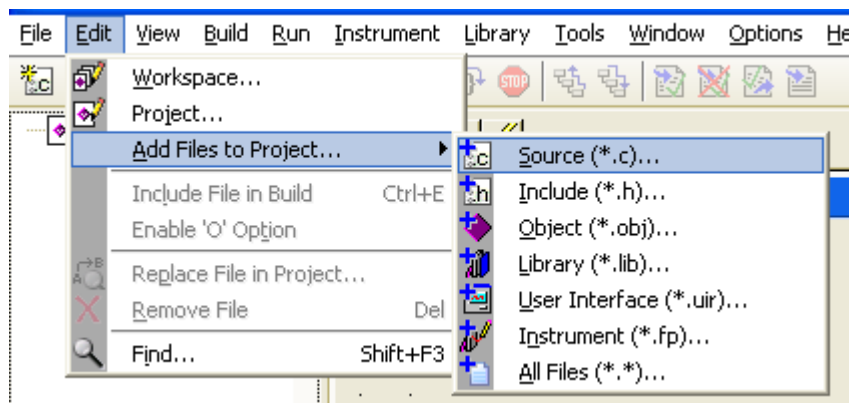


Figura 56. Como añadir los ficheros necesarios en LabWindows.

Una vez se ha creado el proyecto donde se va a trabajar, el siguiente paso es diseñar la interfaz gráfica con el *User Interface Editor* de LabWindows que permite crear interfaces de usuario sin necesidad de generar líneas de código.

Se selecciona el fichero *panel.uir*, automáticamente se abre la ventana *User Interface Editor*, y mediante las cuatro herramientas que posee el editor que muestra la Figura 57, se puede empezar a crear los paneles necesarios para cada proyecto.



Operate Tool. Herramienta de operación.

Edit Tool. Selección de objetos para cambiar su tamaño o posición.

Text Tool. Modificación del texto.

Paint Tool. Paleta de colores.

Figura 57. Herramientas del editor de LabWindows.

En este proyecto se han creado dos paneles, uno para configurar el puerto serie como el de la Figura 58, y otro donde aparecen las mediciones realizadas por el sistema como el que aparece en la Figura 59.

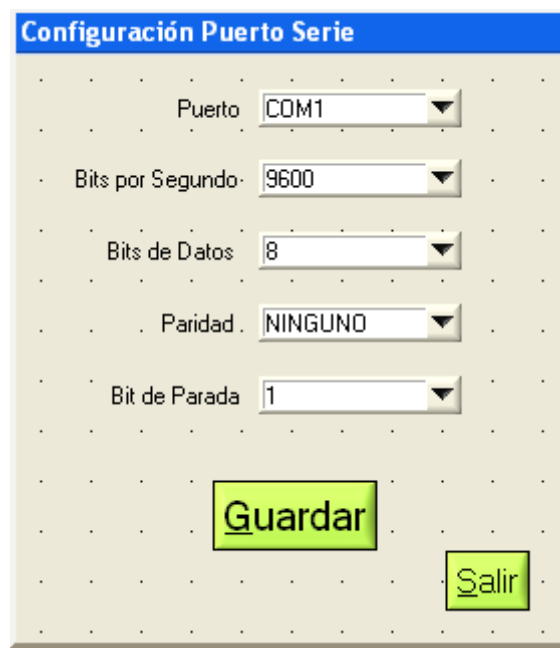


Figura 58. Panel 1 de configuración serie.

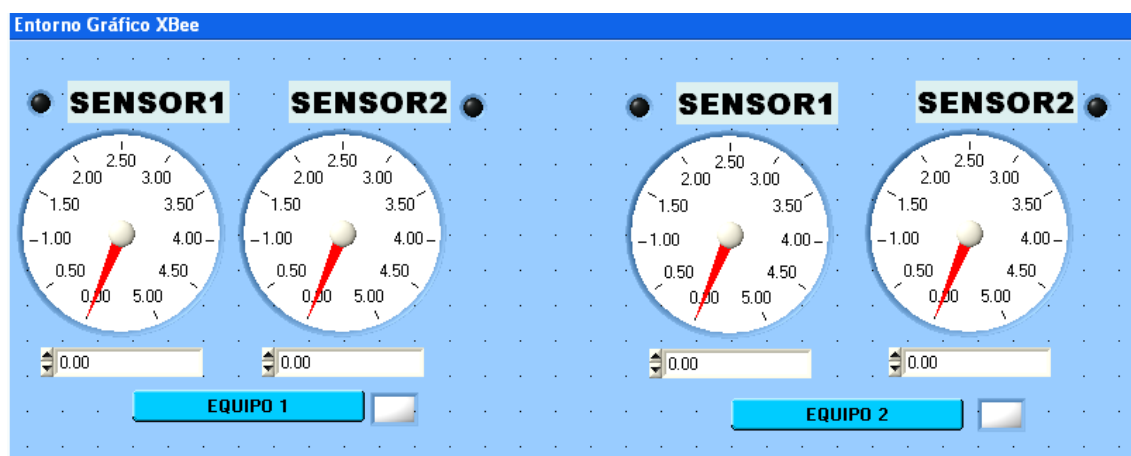


Figura 59. Panel 2 de lectura de medidas.

Para generar los paneles, botones, y distintos elementos que aparecen en cada una de las dos pantallas diseñadas, se selecciona la opción *Create* → *Panel*, *LED*, *Ring*, como se puede ver en la Figura 60.

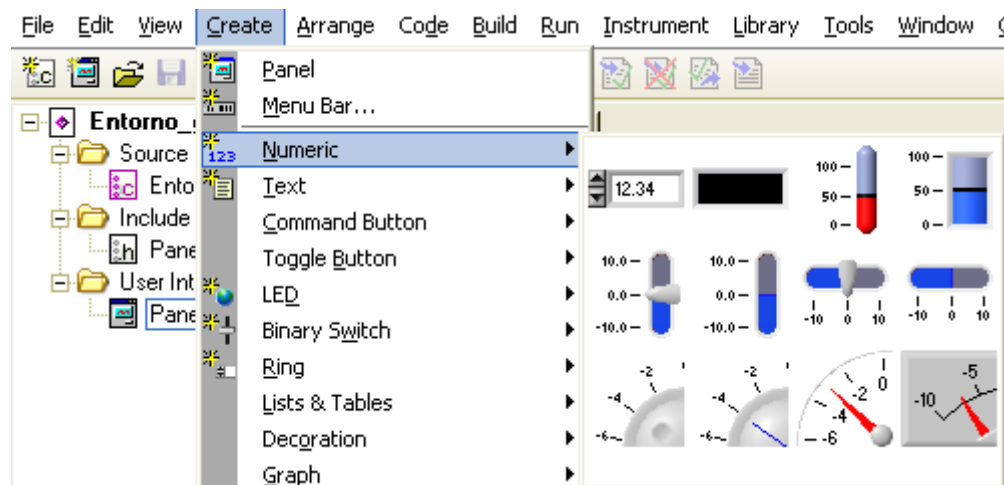


Figura 60. Crear paneles, botones, etc. En LabWindows.

Lo siguiente que se necesita hacer es configurar cada uno de los elementos que se han añadido a cada panel.

Los controles utilizados en este proyecto son:

- Control *Command Button*. En la Figura 61 se puede ver la configuración del botón guardar del panel de la configuración serie.
 - *Constant Name* es el nombre de identificación del botón en el programa.
 - *Callback Function* es el nombre de la función a la que se va a llamar cuando se pulse el botón.
 - *Control Mode*:
 - Normal. El usuario puede operar sobre el control y modificar el programa.
 - *Indicator*. El control se puede modificar por programa pero no por usuario. Los controles LED, text, message, graph son de este tipo.
 - *Hot*. Funciona como el modo anterior, pero se genera un evento cuando el usuario actúa sobre él.

- *Validate*. Funciona como el modo Hot, pero todos los controles del panel se comprueban con respecto a un rango predefinido antes de generar el evento.
- *Label*. Nombre con el que aparece el botón en el diseño final.

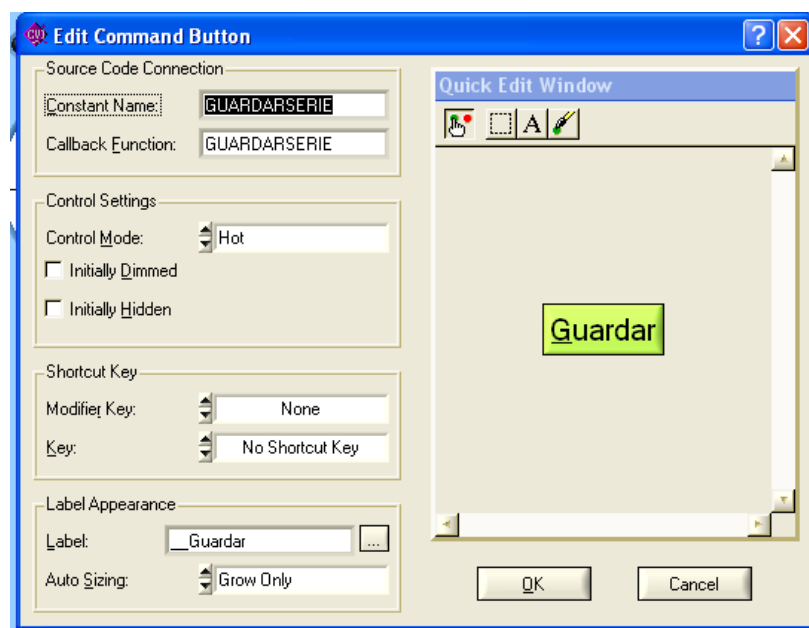


Figura 61. Configuración de un *Command Button*.

- *Control Numeric Gauge*. La configuración de este tipo de control es semejante a la configuración vista anteriormente. La diferencia es que aquí se fija un valor por defecto de la cantidad que se quiere medir y del tipo de dato que se quiere medir mediante el *Control Settings* que aparecen en la Figura 62.

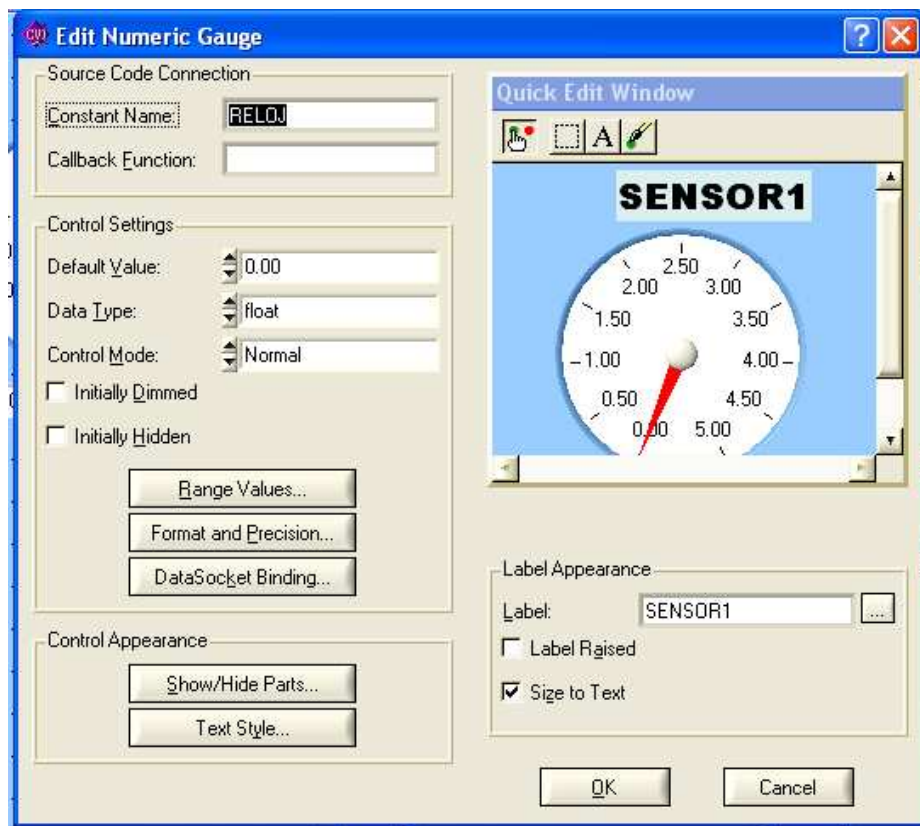


Figura 62. Configuración del control *Numeric Gauge*.

- Control LED. Este control se configura como los casos explicados anteriormente, y el cuadro de diálogo a manejar es el que se puede observar en la Figura 63.

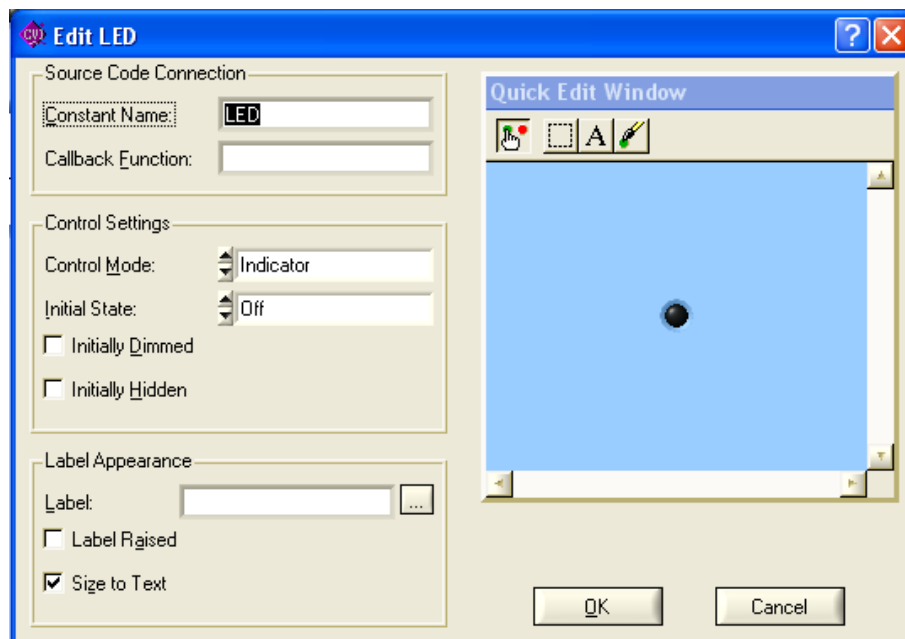


Figura 63. Configuración del control LED.

- Control *Ring*. Se realizan los mismos pasos de configuración que en los casos anteriores a diferencia de que aquí se establecen los valores que puede tener el objeto y el valor por defecto (*Default Index*) como se puede ver en la Figura 64.

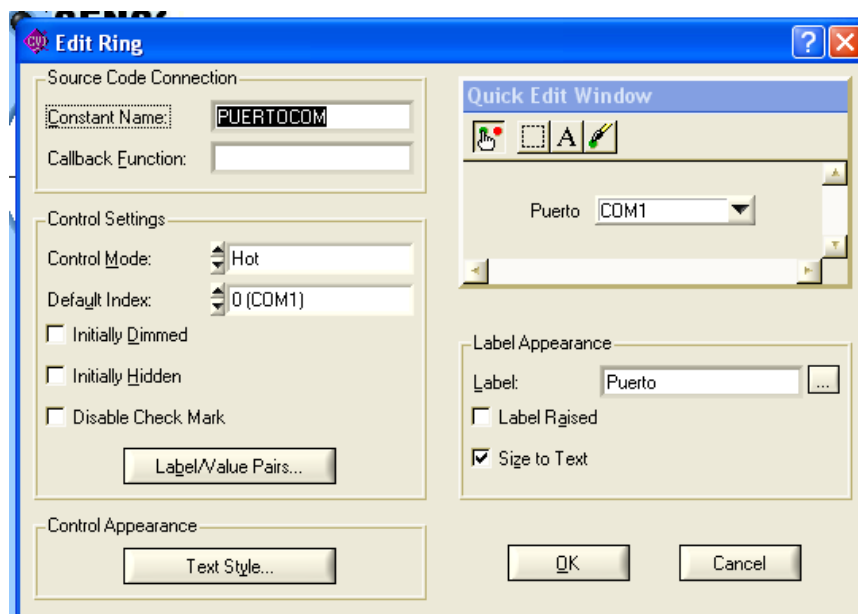


Figura 64. Configuración control *Ring*.

Anexo 2. X-CTU.

Mediante el *software* X-CTU, provisto por Digi es posible programar los módulos XBee, actualizar sus firmwares y realizar todas las pruebas necesarias para verificar el buen funcionamiento de los mismos.

También se puede utilizar el Hyperterminal de Windows para configurar los módulos, pero el X-CTU proporciona una forma más rápida y sencilla de utilización.

Para configurar cada dispositivo XBee se utiliza el Kit Arduino-Diecimila como el de la Figura 65, fabricado por Libelium nombrado a lo largo de este proyecto. La imagen de la izquierda muestra la parte inferior de la placa, a ella va unida la placa donde se encuentra el módulo XBee como se puede ver en la imagen derecha de la Figura 65

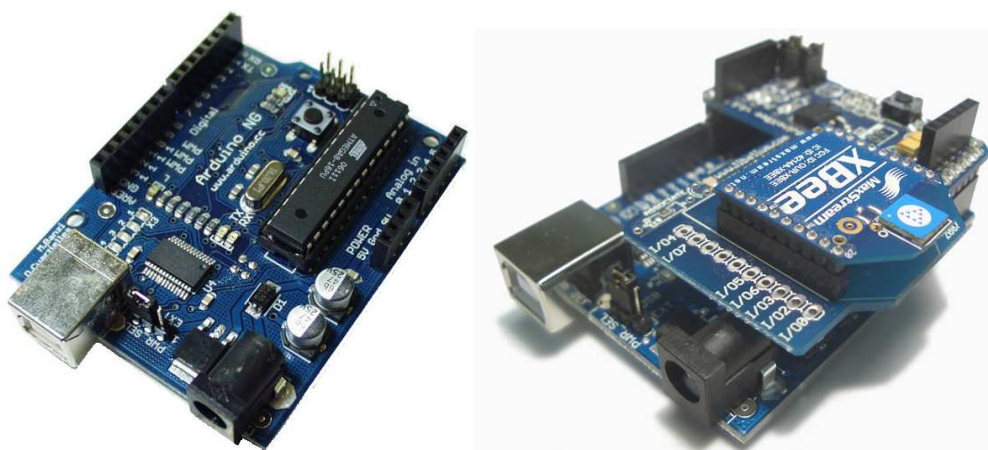


Figura 65. Kit Arduino- Diecimila.[Arduino]

La configuración debe realizarse sin el microprocesador que aparece la imagen izquierda de la Figura 65, en la imagen de la izquierda, y con los *jumpers* superiores e inferiores en modo USB. Antes de abrir el programa X-CTU el dispositivo debe estar conectado al puerto USB del ordenador.

Una vez abierto el X-CTU, en la pestaña inicial de **PC Settings** que muestra la Figura 66, se selecciona el puerto serie (que depende del puerto USB donde se conecte el coordinador al ordenador), se configura la velocidad, 9600 Baud, el control de flujo, el número de bits de datos, la paridad y el número de bits de parada del módulo.

Después se verifica presionando el botón **TEST** apareciendo la información del modelo del XBee que se usa y la versión del *Firmware*

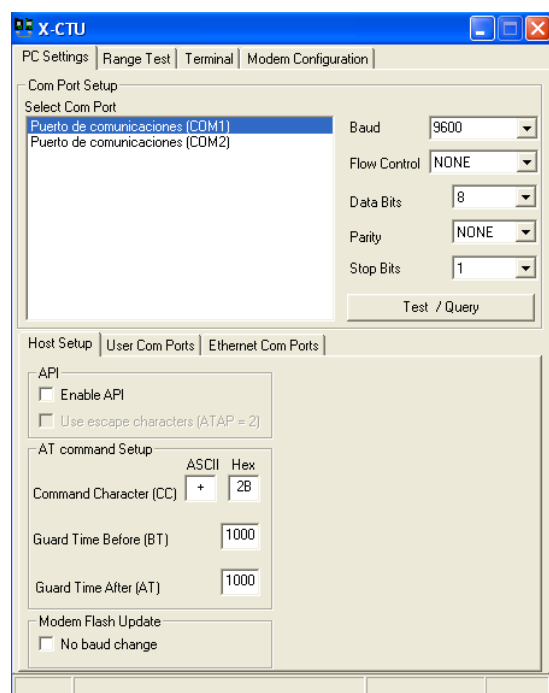


Figura 66. Test de verificación del módulo XBee.

Ahora se pasa a la configuración del dispositivo seleccionando la pestaña *Modem Configuration*, se pulsa el botón *Read* para configurar el dispositivo y aparece la Figura 67, donde se puede ver como se configura cada módulo XBee para que funcionen en una Red *Non Beacon*.

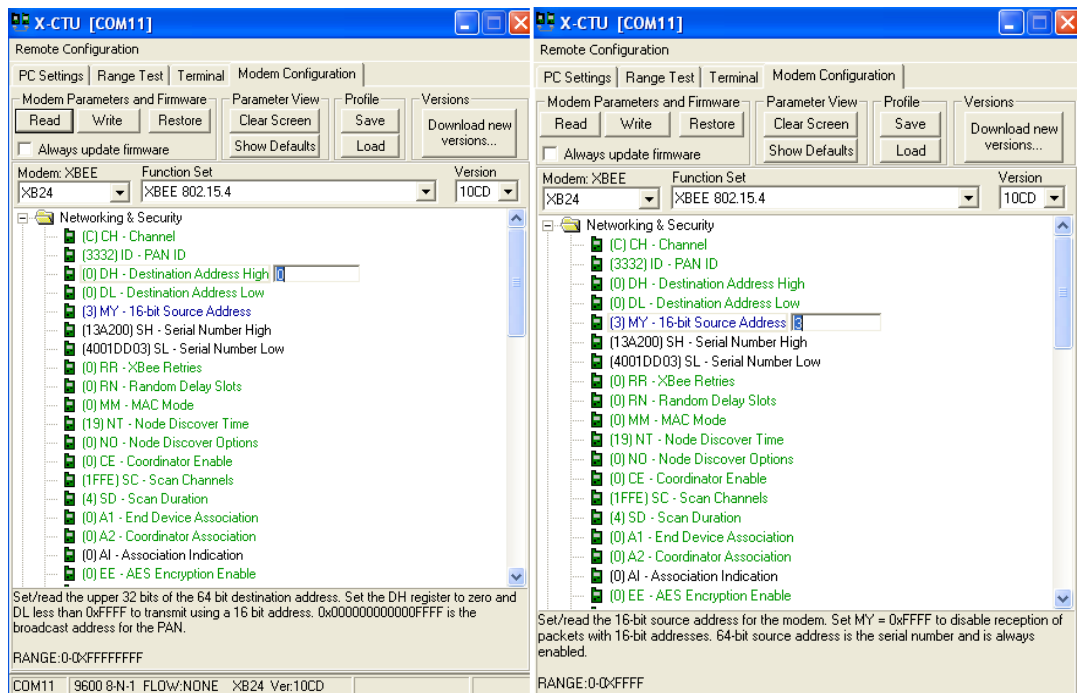


Figura 67. Direcciones de destino y de cada módulo.

Dependiendo de si el módulo es coordinador o *End Device* estos parámetros tendrán un valor diferente. Esto también ocurre con los parámetros seleccionados en la Figura 68, que varían dependiendo si el dispositivo funciona en modo transparente o en modo API, o si es coordinador o *end device*.

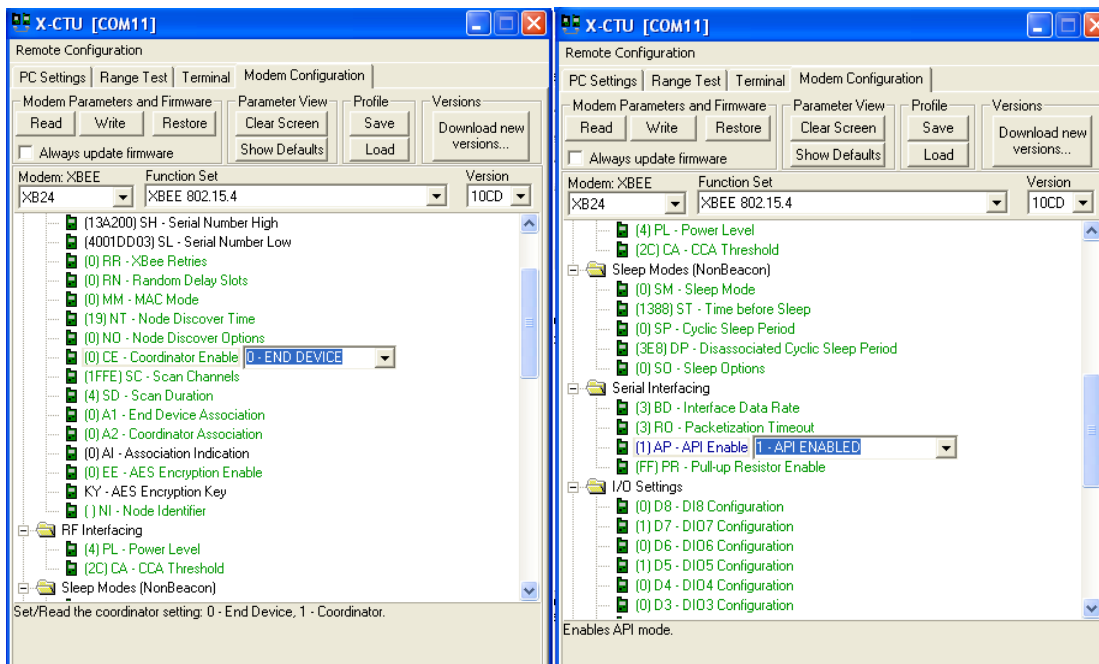


Figura 68. Configuración como *End Device* o como coordinador y Modo de funcionamiento API.

A continuación se muestra la Tabla 28 con los valores de los parámetros usados en cada dispositivo.

	COORDINADOR	END DEVICE 1	END DEVICE 2	END DEVICE 3
Destination Address(DH)(DL)	0	0	0	0
Source Address(MY)	0	1	2	3
Coordinator Enable(CE)	1	0	0	0
Api Enable(AP)	1	1	1	1

Tabla 28. Valores de Configuración para el modo API de cada módulo XBee.

En una configuración en modo transparente, donde sólo existen dos módulos XBee, la configuración anterior no sería válida y habría que modificar los parámetros según la

Tabla 29.



	EMISOR	RECEPTOR
<i>Destination Address(DH)(DL)</i>	40	30
<i>Source Address(MY)</i>	30	40
<i>Api Enable(AP)</i>	1	0

Tabla 29. Valores para una configuración en modo transparente de cada módulo XBee.

Anexo 3. Arduino.

El kit Arduino Diecimila utilizado en este proyecto tiene un Microprocesador atmega168, un chip sencillo que será programado mediante el Entorno de Desarrollo de Arduino (del inglés IDE - *Integrated Development Environment*). Una herramienta que funciona en diferentes plataformas (Windows, Macintosh, Linux) dedicada a la edición y descarga de programas en la placa Arduino.

El sistema está basado en el uso de *Processing* que está hecho en Java. El código que se ejecuta en las placas está basado en las librerías de aplicación *Wiring* API y en las de desarrollo de los integrados AVR. Este es un *software* gratuito que se puede adquirir en la página web de Arduino.

Se crea un nuevo proyecto, se compila, se ejecuta y se carga en el microprocesador mediante los botones de la barra de herramientas que se pueden ver en la Figura 69. Es un *software* sencillo de usar, y todas las funciones necesarias pueden ser encontradas en la página web de Arduino.

[Arduino]

Algunas de las funciones usadas son:

- *Serial.available()* . Obtiene un número entero con el número de bytes disponibles para leer o capturar desde el puerto serie.
- *Serial.Read()* . Lee o captura un byte desde el puerto serie. Siempre que se recibe un byte hay que leerlo.
- *Serial.print()* . Vuelca o envía un número o una cadena de caracteres, al puerto serie. Dicho comando puede tomar diferentes formas, dependiendo de los parámetros que utilicemos para definir el formato de volcado de los números.

- *Serial.begin()*. Inicializa o abre el puerto serie asignándole una velocidad de transmisión de datos (bits por segundo). Dicha función debe ser llamada desde dentro de la estructura o sección *setup()*.

El programa también cuenta con un hyperterminal donde visualizar los resultados obtenidos.



```
/* Blinking LED
 * -----
 *
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, in intervals of 2 seconds. Ideally we use pin 13 on the Arduino
 * board because it has a resistor attached to it, needing only an LED
 *
 * Created 1 June 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 * based on an original by H. Barragan for the Wiring i/o board
 */

int ledPin = 13;          // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Figura 69. Software de Arduino.



Anexo 4. Pliego de condiciones.

El presente proyecto “Red inalámbrica de sensores de fibra óptica de plástico con tecnología ZigBee” se ha realizado en el Grupo de Displays y Aplicaciones Fotónicas del Departamento de Tecnología Electrónica de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid.

El objetivo principal del proyecto ha sido el la implementación de una red sensorial inalámbrica (WSN), basada en el estándar IEEE.802.15.04, y el diseño de una interfaz gráfica que permita la monitorización de las medidas recogidas por los nodos sensores en tiempo real.

La directora del Proyecto ha sido la Dra. Dña. Carmen Vázquez García, Catedrática del Departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid.

Esta línea de investigación, así como el dispositivo realizado y toda la documentación desarrollada, están amparados en el siguiente Pliego de Condiciones:

1. La propiedad intelectual de los estudios realizados en el presente proyecto pertenece por entero a la Universidad Carlos III de Madrid (a partir de ahora, la Universidad)
2. La Universidad se reserva el derecho de la utilización total o parcial del presente Proyecto, bien para su publicación o para su uso en trabajos o proyectos posteriores.
3. En cualquier tipo de reproducción, ya sea para uso particular de la Universidad o para cualquier otra aplicación, se indicará la procedencia, mostrando de manera explícita el nombre del Proyecto, el del ingeniero Ponente, su director y el de la Universidad.
4. Si el proceso pasa la etapa de desarrollo, cualquier modificación que se realice en él deberá ser notificada al Ingeniero Ponente del Proyecto, o al director y, según el criterio de éste, la Universidad decidirá o no la modificación propuesta.
5. En el caso de no aceptar dicha modificación, la Universidad declina toda responsabilidad derivada de la aplicación de la misma.

6. La Universidad tendrá prioridad respecto a otros solicitantes en la elaboración de Proyectos Auxiliares que fuesen necesarios para la posible aplicación industrial que pudiera derivarse del presente Proyecto, siempre y cuando no renuncie de manera explícita a este derecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

Material Utilizado.

Durante las fases de diseño, desarrollo e implementación del prototipo presentado en este proyecto se han utilizado las siguientes herramientas y aplicaciones software:

- Cuatro Kits compuestos de un Arduino Diecimila, y cuatro módulos de comunicación XBee que permiten la transmisión inalámbrica.
- Cuatro emisores ópticos HFBR- 1528.
- Cuatro fotodetectores IF- D91.
- Cuatro cables de fibra óptica de plástico
- Tres Cables USB A-B y un cargador de red.
- Osciloscopio TEKTRONIX, modelo TDS 210.
- Tres fuentes de alimentación AC/DC de ~220 a 9Vcc.
- Mesa óptica TMC, modelo 5085326330 y soportes ópticos.
- Polímetro digital MASTECH, modelo MY-63.
- Ordenador Intel (R) Core™ i3 CPU 5502 @ 3,20 GHz
- Software Entorno de Desarrollo de ARDUINO.



- Software XCTU.
- Software LabWindows de National Instruments.
- Software Microsoft Office Professional Edition 2007.



Anexo 5. Presupuesto.

En el presupuesto que se presenta a continuación, se hace un estudio estimado de los costes de realización del Proyecto, en el que se han incluido los costes relativos al material, diseño y personal.

Descripción	Especificación	Unidades	Coste unitario	Coste total
ARDUINO + XBee	DIECIMILA	4	61 €	244 €
CABLE USB	A – B	4	4 €	16 €
RESISTENCIAS	100 K Ω , 150K Ω , 1M Ω	10	0,05 €	0,5 €
FIBRA ÓPTICA	POF	4,5 m	1 €/m	4,5 €
EMISORES ÓPTICOS	HFBR- 1528	4	20,66 €	82,64 €
FOTODIODOS	IF D91	4	3 €	12 €
			TOTAL	359,64 €



Costes del personal.

Este apartado de costes es el atribuido al trabajo realizado por las personas que han participado en el desarrollo del proyecto. En este proyecto se requiere la colaboración de dos personas: un Ingeniero Industrial y un Secretario.

Para el cálculo de los costes de personal se han supuesto los siguientes salarios brutos anuales:

TRABAJADOR	SUELDO
Ingeniero Industrial	41.151€
Secretario	10.000€

Teniendo en cuenta que en un año de trabajo se computan 1800 horas de trabajo, estos salarios suponen un coste por hora de cada trabajador de:

TRABAJADOR	€/HORA
Ingeniero Industrial	22,86€ /hora
Secretario	5'55€ /hora

Las cargas sobre los salarios son:

- Seguridad Social.....27 %
- Desempleo..... 1,55 %
- Formación Profesional 0,1 %
- Total..... 28,65 %

A partir de los datos anteriores se establece que los gastos personales totales, describiendo el concepto y número de horas de trabajo, son:



CONCEPTO	HORAS	COSTE / HORA	COSTE TOTAL
Diseño y Programación	600	22,86 €/hora	13.716€
Caracterización y Pruebas	200	22,86 €/hora	4.572€
Preparación del documento y redacción	180	5'55€/hora	990 €
			19.278€

Una vez calculados los costes de material y personal obtenemos el presupuesto total del proyecto fin de carrera:

COSTES	PRESUPUESTO
Costes de material	359,64 €
Costes de personal	19.278€
	19.637,64 €

El presupuesto total de este Proyecto Fin de Carrera asciende a: **19.637 Euros, con 64 céntimos de euro.**



Anexo 6. Código fuente.

I. FASE III DE DESARROLLO.

➤ RED SENSORIAL PUNTO A MULTIPUNTO CON 3 *END DEVICE* Y UN COORDINADOR.

• PROGRAMA DE CALIBRACIÓN DE MÓDULOS ARDUINO.

```
/****** MODULOS EMISORES ARDUINO * *****/  
/****** PROGRAMA END DEVICE *****/  
/****** DE CALIBRACIÓN DE MÓDULOS *****/  
/**** AUTOR : Mercedes Chaparro de la Peña *****/
```

```
#include <avr/io.h>
```

```
// DECLARACIÓN DE VARIABLES
```

```
int array[10];
```

```
int incomingByte;
```

```
int,i=0;
```

```
int canal1, canal11,decim5,decim2, decim, decim1;
```

```
int baja,inicio=0;
```



```
int valor[11],modulo;

int analogPin0 = 0; // Pin 0 habilitado como ENTRADA analógica

int analogPin1 = 1; // Pin 1 habilitado como ENTRADA analógica

int analogPin2 = 2; // Pin 2 habilitado como ENTRADA analógica

int analogPin3 =3; // Pin 3 habilitado como ENTRADA analógica

int analogPin4 = 4; // Pin 4 habilitado como ENTRADA analógica

int analogPin5 =5; // Pin 5 habilitado como ENTRADA analógica

int medir=0,a,descomposicion;

int canal, dec, dec1,dec2, dec3,dec4,dec5;

int lecturaPIN0, lectura0,unidadesPIN0,resta0,decimales0,decimalPIN0,compPIN0,n;

int lecturaPIN1, lectura1,unidadesPIN1,resta1,decimales1,decimalPIN1,compPIN1;

int lecturaPIN2, lectura2,unidadesPIN2,resta2,decimales2,decimalPIN2,compPIN2;

int valor0= 9000,valor1= 9000 ,valor2,valor3,valor4,valor5,b,e=0;

int resto= 0xFF,suma,c,d;

byte recibo[30];

unsigned int b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13;

// FUNCIÓN DE INICIALIZACION

void setup()

{

  Serial.begin(9600);

// Inicializacion de variables

  tecla=0;

  valor0= 9000; // Variables usadas para comprobar si la medida ha cambiado

  leerPIN0();// Lectura inicial de cada sensor
```



```
}

// FUNCION PRINCIPAL

void loop()

{   if(Serial.available() > 0)

    {   valor[i]=Serial.read();

        i++;

            if(i==11)

                {i=0;

                    valor0=0;

                    valor1=0;

                }

    }

    b=0;e=0;

    leerPIN0(); // Lectura de cada sensor

    e=0; }

/*FUNCION LECTURA DEL PIN ANALOGICO 0*/

void leerPIN0(void)

{

    int i;

    b2=0;

    delay(3000);

    for(i=0;i<10;i++){

        array[i]=analogRead(analogPin0);

        b2=b2+array[i];
```



```
}  
  
b2=b2/10;  
  
b3=(b2*0.4);  
  
Serial.print("\n ADC ");Serial.print(b2,DEC);  
  
Serial.print("\n V = ");Serial.print(b3,DEC);  
  
Serial.print("\n\n\n ");  
  
}
```

- **PROGRAMA DE CADA DISPOSITIVO *END DEVICE*.**

```
/***** MODULOS EMISORES ARDUINO *****/  
  
/***** PROGRAMA END DEVICE *****/  
  
/***** LEE DOS ENTRADAS ANALÓGICAS DEL *****/  
  
/***** MÓDULO ARDUINO-XBee EMISOR *****/  
  
/***** LO ENVIA AL MÓDULO ARDUINO-XBee RECEPTOR *****/  
  
/**** AUTOR : Mercedes Chaparro de la Peña *****/
```

```
#include <avr/io.h>
```

```
// DECLARACIÓN DE VARIABLES
```

```
int array[10];
```

```
int incomingByte;
```

```
int,i=0;
```



```
int canal1, canal11,decim5,decim2, decim, decim1;

int baja,inicio=0;

int valor[11],modulo;

int analogPin0 = 0; // Pin 0 habilitado como ENTRADA analógica

int analogPin1 = 1; // Pin 1 habilitado como ENTRADA analógica

int analogPin2 = 2; // Pin 2 habilitado como ENTRADA analógica

int analogPin3 =3; // Pin 3 habilitado como ENTRADA analógica

int analogPin4 = 4; // Pin 4 habilitado como ENTRADA analógica

int analogPin5 =5; // Pin 5 habilitado como ENTRADA analógica

int medir=0,a,descomposicion;

int canal, dec, dec1,dec2, dec3,dec4,dec5;

int lecturaPIN0, lectura0,unidadesPIN0,resta0,decimales0,decimalPIN0,compPIN0,n;

int lecturaPIN1, lectura1,unidadesPIN1,resta1,decimales1,decimalPIN1,compPIN1;

int lecturaPIN2, lectura2,unidadesPIN2,resta2,decimales2,decimalPIN2,compPIN2;

int valor0= 9000,valor1= 9000 ,valor2,valor3,valor4,valor5,b,e=0;

int resto= 0xFF,suma,c,d;

byte recibo[30];

unsigned int b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13;

// FUNCIÓN DE INICIALIZACION

void setup()

{

  Serial.begin(9600);

// Inicializacion de variables

  tecla=0;
```



```
valor0= 9000; // Variables usadas para comprobar si la medida ha cambiado

valor1= 9000; // Variables usadas para comprobar si la medida ha cambiado

leerPIN0();// Lectura inicial de cada sensor

leerPIN1();// Lectura inicial de cada sensor

}

// FUNCION PRINCIPAL

void loop()

{   if(Serial.available() > 0)

    {   valor[i]=Serial.read();

        i++;

        if(i==11)

        {i=0;

            valor0=0;

            valor1=0;

        }

    }

    b=0;e=0;

    leerPIN0(); // Lectura de cada sensor

    leerPIN1(); // Lectura de cada sensor

    e=0; }

/*FUNCION LECTURA DEL PIN ANALOGICO 0*/

void leerPIN0(void)

{ int i;
```



```
b2=0;

delay(1000);

for(i=0;i<10;i++){

array[i]=analogRead(analogPin0);

    b2=b2+array[i];

}

b2=b2/10;

b1=(b2*0.4)+(b2/19);

lecturaPIN0=b1/1;

lectura0=lecturaPIN0/1;

if(lectura0<0)

{lectura0=0;}

unidadesPIN0=lectura0/100;

if(lecturaPIN0==191)

{unidadesPIN0=5;

decimalPIN0=0;

e=1;}

if(unidadesPIN0==5)

{unidadesPIN0=5;

decimalPIN0=0;

e=1;}

resta0=unidadesPIN0*100;

decimales0=lectura0-resta0;

decimalPIN0=decimales0/1;
```



```
if(unidadesPIN0==5)

{decimalPIN0=0;}

compPIN0=lecturaPIN0/1;

if(valor0!=compPIN0)

{ a=0;

b=1;

//suma=133;

suma=1+131+0+0+1;

    valor0=lecturaPIN0/1;

    c=resto-(suma+a+e+unidadesPIN0+decimalPIN0);

//ENVIO TRAMA DE DATOS

Serial.write(126);//7E

Serial.write(a);//0

Serial.write((9));//9 BYTES DE ENVIO DE AQUI AL CHECKSUM

Serial.write((1));//API ID 1

Serial.write((131));//FRAME ID =0X83 , XQ SON 16 BITS DE DIRECCIONAM

Serial.write(a);//DL

Serial.write(a);//DL

Serial.write((1));//OPTIONS

Serial.write(a);// canal 0

Serial.write(e);// error, 0 o 1

Serial.write(unidadesPIN0);//medida de 0 a 5

Serial.write(decimalPIN0);// medida de 0 a 5

Serial.write(c);//1+83+1+48+4F+4C+41-FF= 56
```



```
}  
  
}  
  
/*FUNCION LECTURA DEL PIN ANALOGICO 1*/  
  
void leerPIN1(void)  
{int i;  
  
b2=0;  
  
delay(1000);  
  
for(i=0;i<10;i++){  
array[i]=analogRead(analogPin1);  
  
b2=b2+array[i];  
}  
  
b2=b2/10;  
  
b1=(b2*0.4)+(b2/19);  
  
lecturaPIN1=b1/1;  
  
lectura1=lecturaPIN1/1;  
  
if(lectura1<0){lectura1=0;}  
  
unidadesPIN1=lectura1/100;  
  
if(lecturaPIN1==191 || unidadesPIN1==5){unidadesPIN1=5;e=1;}  
  
resta1=unidadesPIN1*100;  
  
decimales1=lectura1-resta1;  
  
decimalPIN1=decimales1/1;  
  
if(unidadesPIN1==5){decimalPIN1=0;}  
  
compPIN1=lecturaPIN1/1;  
  
if(compPIN1!=valor1)
```



```
{  
  
a=1;  
  
b=1;  
  
int cero=0;  
  
suma=133;  
  
    valor1=lecturaPIN1/1;  
  
    c=resto-(suma+a+e+unidadesPIN1+decimalPIN1);  
  
Serial.write(126);//7E  
  
Serial.write(cero);//0  
  
Serial.write((9));//9 BYTES DE ENVIO DE AQUI AL CHECKSUM  
  
Serial.write((1));//API ID 1  
  
Serial.write((131));//FRAME ID =0X83 , XQ SON 16 BITS DE DIRECCIONAM  
  
Serial.write(cero);//DL  
  
Serial.write(cero);//DL  
  
Serial.write((1));//OPTIONS  
  
Serial.write(a);// canal 0  
  
Serial.write(e);// error, 0 o 1  
  
Serial.write(unidadesPIN1);//medida de 0 a 5  
  
Serial.write(decimalPIN1);// medida de 0 a 5  
  
Serial.write(c);//1+83+1+48+4F+4C+41-FF= 56  
  
}  
  
}
```



- **APLICACIÓN GRÁFICA EN LabWindows.**

```

/*****
/*****      MODULOS EMISORES ARDUINO      * *****/
/*****      APLICACIÓN GRÁFICA      *****/
/*****      LEE DOS ENTRADAS ANALÓGICAS DE      *****/
/*****      TRES MÓDULOS ARDUINO-XBee EMISORES      *****/
/*****      MONITORIZA LOS DATOS Y LOS GUARDA EN UN FICHERO TXT      *****/
/*****      AUTOR : Mercedes Chaparro de la Peña      *****/
/*****/

#include <formatio.h>

#include "toolbox.h"

#include <rs232.h>
#include <cvirte.h>
#include <userint.h>
#include <ansi_c.h>
#include <utility.h>
#include <math.h>
#include "codigosASCII.h"
#include "Panel.h"

#define NUMERO_BYTE      13
#define FILE_NAME "data.txt"

//*****/
```

//VARIABLES GLOBALES

//*****

```
static int panelPrincipal, panelConfSerie, panelPicture;
```

```
int chivato_confserie = 0;           //Nos indica si se ha entrado a configurar
                                     //el puerto de comunicaciones serie
serie RS232                          //para mandar un mensaje de aviso
```

```
int chivato_modoauto = 0;
```

```
int chivato_espera = 0;
```

```
int contador;  
  
//Variables del puerto serie  
unsigned char mi_puertocom = 0;  
  
int mi_bitsporsegundo;  
  
unsigned char mi_paridad;  
  
unsigned char mi_bitsdedatos;  
  
unsigned char mi_bitdeparada;  
  
int time2;
```

```
int error,valor=4; //Variable donde almacenar la respuesta de las
funciones
```

```
int opcion; //Variable donde se almacena la opción del mensaje de aviso
```

```
char codigo = 0;          //Variable de petición de información adicional código 0
float  compara[50];
```



```
unsigned char dato_tx[11];    //Dato a enviar por el puerto serie
unsigned char dato_rx[49];    //Dato recibido por el puerto serie
float dato_rhx[50];          //Dato recibido por el puerto serie
int dato_rxhh[13];
unsigned char i = 0;          //Variable indices
unsigned char j = 0;
unsigned char k = 0;
char dato_rx fallo[100]="sobrepasada la medida ";
char cambio_base[12];
int contador;
char salida[7]="HOLA";
int num=1,p=0,a=0;
char buffer2[100];

int fh2;

char fh3[8];

static FILE *fh1;

char *titulo= " TIEMPO\tEQUIPO\tSENSOR\tLECTURA\tALARMA\n";

char buf2[100];

//*****
//FUNCIÓN PRINCIPAL
//*****
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;    /* out of memory */

    fh2=OpenFile (FILE_NAME, VAL_WRITE_ONLY, VAL_OPEN_AS_IS, VAL_ASCII );
    WriteFile(fh2,titulo, strlen(titulo));
```



CloseFile (fh2);

```
//Cargamos todos los paneles
//panelPicture = LoadPanel (0, "Panel.uir", INICIO);
panelPrincipal = LoadPanel (0, "Panel.uir", PRINCIPAL);
panelConfSerie = LoadPanel (0, "Panel.uir", SERIE);
```

```
//Mostramos el panel "Principal"
DisplayPanel (panelPrincipal);
RunUserInterface ();
return 0;
```

```
}
```

```
//*****
```

```
//FUNCIÓN BOTON testeo
```

```
//*****
```

```
int CVICALLBACK testeo (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2)
```

```
{
```

```
    switch (event)
```

```
    {
```

```
        case EVENT_COMMIT:
```

```
            DefaultPanel (panelPrincipal);
```

```
            compara[1]=0;
```

```
            compara[11]=0;
```

```
            compara[2]=0;
```

```
            compara[22]=0;
```

```
            compara[3]=0;
```

```
            compara[33]=0;
```



```
a=1;
dato_tx[0]=126;
dato_tx[1]=0;
dato_tx[2]=7;
dato_tx[3]=1;
dato_tx[4]=131;
dato_tx[5]=0;
dato_tx[6]=a;
dato_tx[7]=1;
dato_tx[8]=0;
dato_tx[9]=0;
dato_tx[10]=255-(1+131+0+a+1+0+0);

for(a=1;a<=3;a++)
{
    dato_tx[6]=a;

ComWrt(mi_puertocom, dato_tx, 11);
    } a=0;
    break;
}
return 0;
}

//*****
//SUBROUTINA DE ATENCIÓN AL TEMPORIZADOR DE LOS LEDS
//*****

void CVICALLBACK temporizador (void *callbackData)
{
    SetCtrlVal (panelPrincipal, PRINCIPAL_LED,0);
```



```
}  
void CVICALLBACK temporizador2 (void *callbackData)  
{  
    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_2,0);  
}  
void CVICALLBACK temporizador3 (void *callbackData)  
{  
    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_3,0);  
}  
void CVICALLBACK temporizador4 (void *callbackData)  
{  
    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_4,0);  
}  
void CVICALLBACK temporizador5 (void *callbackData)  
{  
    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_5,0);  
}  
void CVICALLBACK temporizador6 (void *callbackData)  
{  
    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_6,0);  
}  
  
//*****  
//FUNCIÓN QUE ANALIZA LOS DATOS  
//*****  
  
void analizar_dato()  
{  
    dato_rhx[1]=(dato_rx[10]);  
    if(dato_rhx[1]<=0)
```



```
{  
  
    dato_rxh[1]=0;  
  
}
```

```
dato_rxh[0]=((dato_rx[11])/100.)+dato_rxh[1];
```

```
if(dato_rxh[0]<=0)  
{  
  
    dato_rxh[0]=0;  
  
}  
  
if(dato_rxh[0]>5)  
{  
  
    dato_rxh[0]=5;  
  
}
```

```
dato_rx[20]=dato_rx[11]/10;
```

```
dato_rx[21]=dato_rx[11]-(dato_rx[20]*10);
```

```
buf2[0]='\t';
```

```
buf2[1]=' ';
```

```
buf2[2]=' ';
```

```
buf2[3]=dato_rx[5]+48;
```

```
buf2[4]='\t';
```

```
buf2[5]=' ';
```

```
buf2[6]=' ';
```

```
buf2[7]=dato_rx[8]+48;
```



```
buf2[8]='\t' ;
buf2[9]=' ' ;
buf2[10]=dato_rxh[1]+48;
buf2[11]=',';
buf2[12]=dato_rx[20]+48;
buf2[13]=dato_rx[21]+48;
buf2[14]='\t';
buf2[15]=' ' ;
buf2[16]=' ' ;
buf2[17]=dato_rx[9]+48;
buf2[18]='\n';

    fh2=OpenFile (FILE_NAME, VAL_WRITE_ONLY, VAL_APPEND, VAL_ASCII );
WriteFile(fh2,TimeStr(), strlen(TimeStr()));

WriteFile(fh2,buf2, strlen(buf2));

CloseFile (fh2);

// REPRESENTACIÓN DE DATOS POR PANTALLA

/*****DATOS DEL EQUIPO 1*****/

if(dato_rx[5]==1)
{

    if(dato_rx[8]==0)
    {

        if(dato_rxh[0]!=compara[1])
```



```
        {
            SetCtrlVal (panelPrincipal, PRINCIPAL_LED,1);
            PostDelayedCall (temporizador, 0, 4);
            SetCtrlVal (panelPrincipal, PRINCIPAL_RELOJ, dato_rxh[0]);
            compara[1]=dato_rxh[0];
        }
    }

    if(dato_rx[8]==1)
    {
        if(dato_rxh[0]!=compara[11])
        {
            SetCtrlVal (panelPrincipal, PRINCIPAL_LED_2,1);
            PostDelayedCall (temporizador2, 0, 4);
            SetCtrlVal (panelPrincipal, PRINCIPAL_RELOJ_2, dato_rxh[0]);
            compara[11]=dato_rxh[0];
        }
    }

    if(dato_rx[9]==1)
    {
        SetCtrlVal (panelPrincipal, PRINCIPAL_LED_7,1);
    }

    if(dato_rx[9]==0)
    {
        SetCtrlVal (panelPrincipal, PRINCIPAL_LED_7,0);
    }
}
```



}

}

/***DATOS DEL EQUIPO 2*****/**

if(dato_rx[5]==2)

{

if(dato_rx[8]==0)

{

if(dato_rhx[0]!=compara[2])

{

SetCtrlVal (panelPrincipal, PRINCIPAL_LED_3,1);

PostDelayedCall (temporizador3, 0, 4);

SetCtrlVal (panelPrincipal, PRINCIPAL_RELOJ_3, dato_rhx[0]);

compara[2]=dato_rhx[0];

}

}

if(dato_rx[8]==1)

{

if(dato_rhx[0]!=compara[22])

{

SetCtrlVal (panelPrincipal, PRINCIPAL_LED_4,1);

PostDelayedCall (temporizador4, 0, 4) ;

SetCtrlVal (panelPrincipal, PRINCIPAL_RELOJ_4, dato_rhx[0]);

compara[22]=dato_rhx[0];

}



```
    }

    if(dato_rx[9]==1)
    {
        SetCtrlVal (panelPrincipal, PRINCIPAL_LED_8,1);
    }

    if(dato_rx[9]==0)
    {
        SetCtrlVal (panelPrincipal, PRINCIPAL_LED_8,0);
    }
}

/*****DATOS DEL EQUIPO 3*****/

if(dato_rx[5]==3)
{
    if(dato_rx[8]==0)
    {
        if(dato_rhx[0]!=compara[3])
        {
            SetCtrlVal (panelPrincipal, PRINCIPAL_LED_5,1);
            PostDelayedCall (temporizador5, 0, 4) ;
            SetCtrlVal (panelPrincipal, PRINCIPAL_RELOJ_5, dato_rhx[0]);

            compara[3]=dato_rhx[0];
        }
    }

    if(dato_rx[8]==1)
    {
```



```
                                if(dato_rhx[0]!=compara[33])
                                {
                                    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_6,1);
                                    PostDelayedCall (temporizador6, 0, 4);
                                    SetCtrlVal (panelPrincipal, PRINCIPAL_RELOJ_6, dato_rhx[0]);

                                compara[33]=dato_rhx[0];
                                }
                                }

                                if(dato_rx[9]==1)
                                {
                                    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_9,1);
                                }
                                if(dato_rx[9]==0)
                                {
                                    SetCtrlVal (panelPrincipal, PRINCIPAL_LED_9,0);
                                }

                                }

                                /*****/

                                }

                                /***/
```



//SUBROUTINA DE ATENCIÓN AL PUERTO SERIE POR INTERRUPCIÓN

//*****

void CVICALLBACK ATENDER_PUERTO_SERIE (int COMPort, int eventMask,

void *callbackData){

 //printf("3");

 if (eventMask & LWRS_RECEIVE){

 ComRd (mi_puertocom, dato_rx, NUMERO_BYTE);

 FlushInQ (mi_puertocom); //printf("4");

 analizar_dato();

 }

}

//*****

//FUNCIÓN BOTON SALIR DEL PANEL PRINCIPAL

//*****

int CVICALLBACK SALIR (int panel, int control, int event,

void *callbackData, int eventData1, int eventData2)

{

 switch (event)

 {

 case EVENT_COMMIT:

 //Cerramos el puerto serie si ha sido abierto

 if(mi_puertocom != 0)

 CloseCom (mi_puertocom);

 QuitUserInterface (0);



```
        break;

    }

    return 0;
}

//*****
//FUNCIÓN BOTON CONFIGURAR EL PUERTO SERIE DEL PANEL PRINCIPAL
//*****

int CVICALLBACK CONFSERIE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            //Abrir el panel de configuración del puerto serie
            if ((panelConfSerie = LoadPanel (0, "Panel.uir", SERIE)) < 0)
                return -1;

            DisplayPanel (panelConfSerie);
            compara[1]=0;
            compara[11]=0;
            compara[2]=0;
            compara[22]=0;
            compara[3]=0;
            compara[33]=0;
            RunUserInterface ();
```



```
        break;

    }

    return 0;

}

//*****
//FUNCIÓN BOTON SALIR DEL PANEL SERIE
//*****

int CVICALLBACK SALIRSERIE (int panel, int control, int event,
                           void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:

            //Cerramos el panel de 'serie'
            DiscardPanel(panelConfSerie);

            break;

    }

    return 0;
}

//*****
//FUNCIÓN BOTON GUARDAR CONFIGURACION DEL PANEL SERIE
//*****

int CVICALLBACK GUARDARSERIE (int panel, int control, int event,
                              void *callbackData, int eventData1, int eventData2)
{

```



```
int notifyCount = NUMERO_BYTE;    //En esta variable indicamos el número de byte que
                                   //hay que recibir para producir una llamada a
la
                                   //función "ATENDER_PUERTO_SERIE" por el
flag LWRS_RECEIVE
                                   //No utilizado

int eventChar = 126;    //Caracter que se espera (10=LF) recibir para provocar una llamada
                                   //a la función "ATENDER_PUERTO_SERIE" por
el flag LWRS_RXFLAG

//int eventMask = LWRS_RXFLAG;
int eventMask =LWRS_RECEIVE;    //El evento que nos provoca la llamada a la función
                                   //"ATENDER_PUERTO_SERIE"
es por recepción de un carcter

switch (event)
{
    case EVENT_COMMIT:

        //Copiamos el valor de los botones de configuración del puerto serie
        GetCtrlVal (panelConfSerie, SERIE_PUERTOCOM, &mi_puertocom);
        GetCtrlVal (panelConfSerie, SERIE_BITSPORSEGUNDO, &mi_bitsporsegundo);
        GetCtrlVal (panelConfSerie, SERIE_PARIDAD, &mi_paridad);
        GetCtrlVal (panelConfSerie, SERIE_BITSDEDATOS, &mi_bitsdedatos);
        GetCtrlVal (panelConfSerie, SERIE_BITDEPARADA, &mi_bitdeparada);

        //Abrimos el puerto serie
```



```
if(mi_puertocom == 1)
    OpenComConfig (1, "COM1", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);
```

```
if(mi_puertocom == 2)
    OpenComConfig (2, "COM2", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);
```

```
if(mi_puertocom == 3)
    OpenComConfig (3, "COM3", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);
```

```
if(mi_puertocom == 4)
    OpenComConfig (4, "COM4", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);
```

```
if(mi_puertocom == 5)
    OpenComConfig (5, "COM5", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);
```

```
if(mi_puertocom == 6)
    OpenComConfig (6, "COM6", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);
```



```
if(mi_puertocom == 7)
    OpenComConfig (7, "COM7", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);

if(mi_puertocom == 8)
    OpenComConfig (8, "COM8", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);

if(mi_puertocom == 9)
    OpenComConfig (9, "COM9", mi_bitsporsegundo,
                    mi_paridad, mi_bitsdedatos,
mi_bitdeparada, 512, 512);

//Sin control de flujo por hardware
SetCTSMode (mi_puertocom, LWRS_HWHANDSHAKE_OFF);
//Sin control de flujo por XON/XOFF
SetXMode (mi_puertocom, 0);
    notifyCount=1;      // printf("1");
//Activamos la llamada a la función "ATENDER_PUERTO_SERIE"
//cuando haya una respuesta en el puerto COM (recepción de un caracter)
InstallComCallback (mi_puertocom, LWRS_RECEIVE, notifyCount,
                    eventChar, ATENDER_PUERTO_SERIE, NULL); //printf("2");
FlushInQ (mi_puertocom);
    //printf("\n");

//Cerramos la ventana
DiscardPanel(panelConfSerie);
```



```
        break;
    }
    return 0;
}
```